Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Evaluating the Bipartite Graph Layout for Network Visualization

Nathan Schiele

Course of Study:	INFOTECH
Examiner:	Prof. Dr. Daniel Weiskopf
Supervisor:	Moataz Abdelaal M.Sc., Katrin Angerbauer M.Sc., Cristina Morariu M.Sc.
Commenced:	April 27, 2020
Completed:	October 27, 2020

Abstract

There have been many studies on network visualization efficacy. This work expands on previous studies examining the efficacies of the common visualizations of Node-Link and Adjacency Matrix visualizations. We add a third visualization, Bipartite Layout, and examine the efficacy of this visualization compared to those examined by previous studies. Our study was performed on the crowdsourcing platform, Mechanical Turk, on a total of 72 participants with varying experiences with network visualization.

Overall, we were able to largely confirm the results of previous visualization efficacy studies. We found that the Bipartite Layout mostly shares task-specific efficacy with the Adjacency Matrix visualization. We also found that the effect of increasing network size on task completion accuracy on the Bipartite Layout appears to be similar to that of the Adjacency Matrix. Similarly, the effect due to increasing network density is comparable to that of the Node-Link visualization.

We found that Bipartite Layouts tend to perform better than other visualizations on large, sparse graphs. We also find that interactivity has a significant effect on network visualization efficacy. Further study will be needed to see which interactive elements cause which effects, and to refine further our understanding of the efficacies of Bipartite Layout.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

This chapter introduces this work, starting with the motivation behind the thesis. We move onto our objective and a list of our contributions. Finally, we close with an overview of the structure of this thesis as a whole.

1.1 Motivation

Data wholly defines the contemporary age. A single visit to any website results in dozens of tracking cookies, collecting all manner of data from the amount of time spent on a page, the location of the mouse, what is clicked on, and more. A wrist-worn fitness tracker can produce dozens of data points every second. We have developed countless methods for collecting vast amounts of data, and we will likely continue to develop even more nuanced methods for collecting even further amounts of data.

All of this information creates several other problems that need to be solved. One of these problems is how to visualize these data to make it understandable. Providing raw data is useless to most people, and understanding optimal methods for conveying the information contained within data is of utmost importance in a world that is increasingly defined by such data. Studies into the most effective methods for sharing information are of critical importance in an age where both data and the visualization of that data can inundate anyone. Knowing the efficacy of different visualization methodologies and the effects of different properties on that efficacy are necessary to create effective visualizations for sharing these data.

Figure 1.1 shown an example of a road network. Straight lines are drawn between cities if there is a direct highway connection between those two cities. These lines are drawn between every pair of cities with a direct highway connection; when all these lines are shown together, we have a network. This concept can be generalized, as is explained in Chapter 2.

Networks are a group or system of interconnected entities; these data are widespread in the modern age. Data with geographic elements are commonly represented as a network; additionally, most relational data is also commonly represented in the form of a network. As network visualizations are becoming a more common method for information visualization, study into the efficacy of different network visualization techniques increases in importance to ensure that these network visualizations are most effective.

Ghoniem *et al.* wrote the seminal paper in the field of network visualization efficacy (NVE). This study compared two basic methodologies for visualizing networks, Node-Link (NL) and Adjacency Matrix (AM), explicit examples of which can be found in Figures 3.3 and 3.4 respectively. The findings were fairly surprising and did challenge conventional wisdom regarding best practices for visualizing information. Previously, the accepted convention was the lack of intuitiveness of the AM

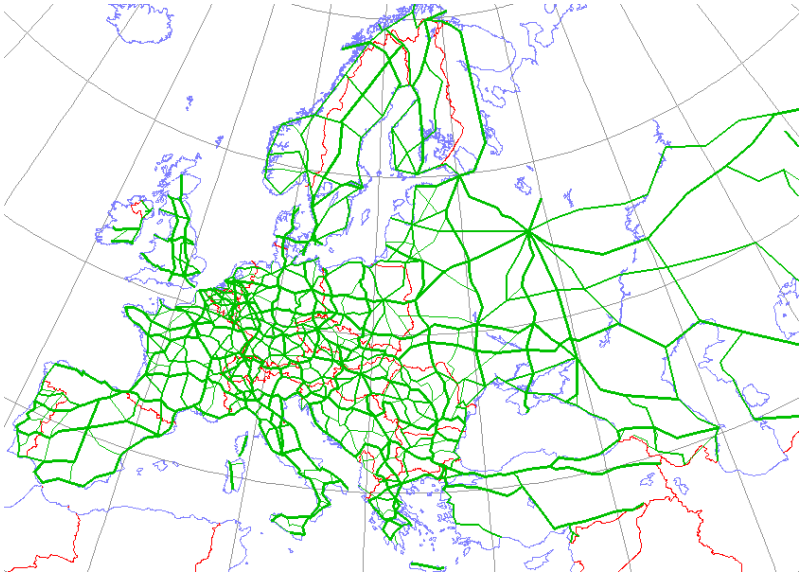


Figure 1.1: An example of a highway network in Europe, highways are shown as straight lines connecting cities [Wik08].

visualization, as will be discussed in Section 3.3, would preclude any understanding or usefulness of the AM visualization. Further studies on network visualizations have clarified for which tasks certain visualizations are effective, and how different properties of network visualizations affect efficacy.

We endeavor to expand our understanding of the efficacy of network visualizations. Specifically, we seek to compare three network representations: NL, AM, and a Bipartite layout (BP); additionally, we examine visualization efficacy without interactivity. Bipartite layouts, as is carefully explained in Section 3.4, have properties of both AM and NL visualizations. It remains to be seen whether this combination of properties is beneficial to the BP visualization's efficacy.

1.2 Objective

This work aims to determine the Bipartite graph layout's efficacy compared to NL and AM graph representations. We will accomplish this by conducting a user study where participants will perform various tasks using one of three different visualization techniques. We will then compare the accuracy and response times across different tasks. Additionally, we will use networks of varying sizes and densities to measure the effects those properties have on the BP visualization compared to the AM and NL visualizations.

There has already been substantial research into network visualization efficacy. Previous studies have primarily focused on the comparison of AM and NL visualizations. To the best of our knowledge, no study to date has researched the efficacy of BP visualizations. Additionally, all

modern NVE studies have implemented various interactive elements in their efficacy assessments. We deliberately do not include interactivity to compare the effects of including interactive elements on network efficacy.

As it compares to NL and AM visualization efficacy, the tasks we use to evaluate BP visualization efficacy have been used before. While we seek to evaluate the efficacy of a visualization, BP, that has not been evaluated in this manner previously, we seek to do so within a framework that has been established by previous works. As such, we do not use novel tasks or a novel approach to evaluating efficacy. This work's novelty is in the new visualization, BP, and the removal of interactivity in the efficacy assessment.

1.3 Contributions

This thesis makes the following contributions:

- Evaluation BP visualization efficacy with respect to different network analysis tasks
- Confirming the previous findings of visualization efficacy between Adjacency Matrix and Node-Link visualizations found in previous NVE studies
- Reviewing the state-of-the-art of NVE studies
- Comparing different network generation models with respect to their use in NVE studies

1.4 Structure

The following chapter will cover a broad overview of the mathematics of graphs to provide a basic understanding of the mathematics necessary to understand subsequent concepts in this thesis. Following that, Chapter 3 is about the visualizations themselves. This chapter builds upon the mathematics introduced in the Chapter 2, and delves into the specifics of how the different network visualization visualize information. In Chapter 4, we discuss the previous studies into network visualization efficacy. This discussion covers the decisions made in previous studies similar to the one we conducted and sets up the discussion of how we designed our study.

After Chapter 4, we have a chapter about the data used in our efficacy study's network visualizations Deciding what data to use was a massive consideration and has wide-ranging impacts on efficacy studies in general, and this discussion takes place in Chapter 5. Following that, we introduce the Study Design chapter, Chapter 6. This chapter discusses the specifics of how we designed and conducted our study and the reasoning behind why we made the decisions that we made. Subsequently, Chapter 7 is about the results we acquired from running the study, and Chapter 8 is the discussion and analysis of those results. Finally, we conclude with a chapter stating our conclusions as well as suggestions for future work.

2 Mathematical Modeling

This chapter provides an overview of the general subject matter and its importance to modern computer science, data science, and the public at large. This chapter will cover a basic overview of what Graph Theory entails, and where graphs are commonly found. We then conclude with a broad discussion of general properties to set up a discussion in the subsequent chapters.

2.1 History of Graph Theory

Graph Theory is a comparatively new field of mathematics. The common story for the origin of the field is Leonhard Euler solving the Bridges of Konigsberg puzzle. To set the puzzle, Konigsberg is a city that lies on the river Pregel. The river splits in the center of the city to form an island, and the river is bridged in seven places to enable the convenient crossing of the river. See figure 2.1 for a visual representation of this puzzle. The puzzle is to find a way to cross every bridge in Konigsberg without crossing a single bridge more than once. The puzzle famously does not have a solution, but the challenge is not that there is no solution; the challenge is proving that there is no solution. Euler first presented his work in 1736, which for many marks the year Graph Theory was born [BLLW76]. Euler first published his proof concerning the solution, or lack thereof, to the Bridges of Konigsberg problem in his 1741 work, "Solutio Problematis ad Geometriam Situs Pertinentis" [Eul41]. The solution from Euler as translated by Biggs *et al.* is as follows,

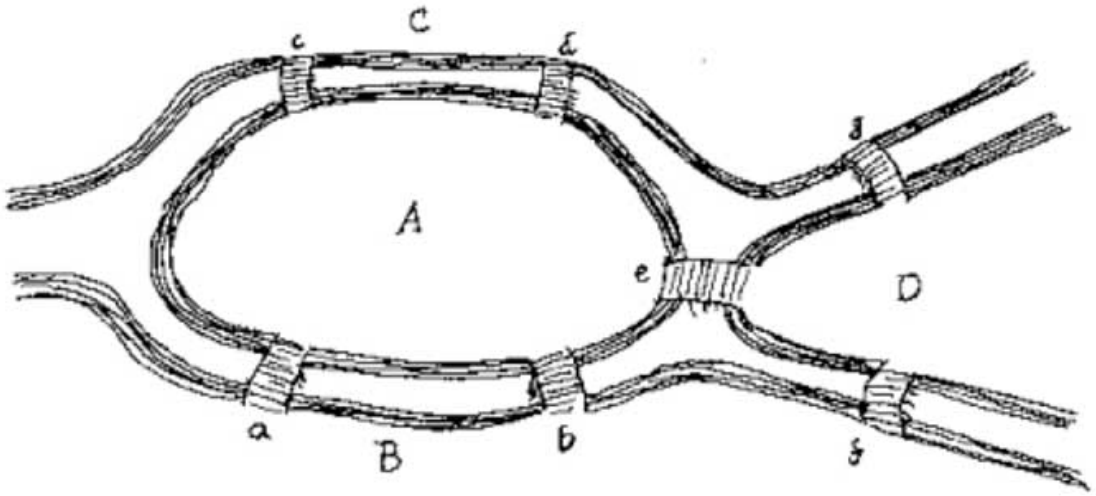


Figure 2.1: Diagram from Euler's work on the Seven Bridges of Konigsberg puzzle [Eul41].

"

- If there are more than two areas to which an odd number of bridges lead then such a journey is impossible.
- If, however, the number of bridges is odd for exactly two areas, then the journey is possible if it starts in either of these areas.
- If, finally, there are no areas to which an odd number of bridges leads, then the required journey can be accomplished starring from any area

With these rules, the given problem can always be solved. " [BLLW76; Eul41].

And given that the problem has four such areas with an odd number of bridges, there is no solution.

From this problem, a new field of mathematics formed. Johann Listing named the study of position, "Topology" in 1848, a field closely related to graph theory. The value of graph theory spans across fields, from theoretical models in mathematics to social network models in psychology to data models in computer science. Critically, graph theory's primary utility is to model new problems and give us the tools to solve them. [LBM78]

2.2 Mathematics of Graphs

This section focuses on the fundamentals of the mathematics used in graph theory.

2.2.1 Defining Graphs

Fundamentally, a graph is a collection of two sets. The first set, V, is a collection of nodes or vertices in the graph. The section set, E, is a collection of all the edges in the graph. An edge is defined as such:

$$e \in E : e = (v, w) \text{ where } v \in V \text{ and } w \in V \tag{2.1}$$

A graph is thus usually denoted $G(V, E)$. The number of nodes in a graph is the cardinality of the set of vertices, $|V|$, and the total number of edges in a graph is the cardinality of the edge set, $|E|$ [Wil96]. We define density as the number of edges divided by the number of vertices, $\frac{|E|}{|V|}$; however, this is not necessarily an accepted convention in graph theory. Section 4.2.3 contains further discussion of how to calculate network density.

Other properties can be used to define graphs, such as edge weights, where edges are given an individual weighting term, and edge multiplicity, where multiple edges can exist between the same pair of nodes.

2.2.2 Graph Directionality

The most common distinguishing characteristic between two types of networks is their directionality. Undirected networks are such that edges just connect two nodes; there is no direction assigned to this connection. In an undirected network, the edge (i, j) between node i and node j just connotes that the two nodes are connected. As such, the edge (j, i) in an undirected network is identical to the edge (i, j). To put shortly, $(i, j) = (j, i)$

In a directed network, this is not the case. An edge in a directed network specifically defines a directional relationship. As such, (i, j) is not the same as (j, i). Put another way, edges in undirected graphs are scalar values while edges in directed graphs are vector values; shortly, $(i, j) \neq (j, i)$.

2.2.3 Node Degree

The degree of a node in an indirected graph is simply the number of edges connected to that node, that is to say,

$$\deg(v \in V) = |\{\in E : e = (w, v) \text{ or } (v, w) \text{ where } w \in V\}| \tag{2.2}$$

For directed graphs, it is not possible to have a single measure of degree, given that directed graphs have two types of edges within the context of a single node, incoming and outgoing edges. Instead we must separate degrees into two separate measures, based on the type of edge,

$$\text{in-degree}(v \in V) = |\{\in E : e = (w, v) \text{ where } w \in V\}| \tag{2.3}$$

$$\text{out-degree}(v \in V) = |\{\in E : e = (v, w) \text{ where } w \in V\}| \tag{2.4}$$

3 Network Visualizations

All network visualization efficacy studies have effectively examined two basic visualizations, the node-link (NL) visualization and the adjacency matrix (AM) visualization. These visualizations have long been considered as the fundamental methods to illustrate networks. As was discussed in Chapter 1, we examine the Bipartite (BP) visualization in addition to NL and AM visualizations. This chapter will explore these three visualizations in-depth and discuss the visual elements and interactivity common to all visualizations.

3.1 Examples of Network Visualizations

Networks are a common feature in data visualizations and can be found in many different contexts. The two examples provided here help showcase the utility of network visualizations and why study into network visualization efficacy is necessary. Both networks shown here are using the NL visualization method.

3.1.1 Airline Network

An excellent example of a network is one representing where airlines fly. In most seatback magazines, one can find an airline advertisement containing a network visualization, usually an NL visualization, of which airports the airline flies to. This advertisement usually intends to inspire future travel with the airline, meaning understandability is critical to this network representation.

Figure 3.1 contains a representation from S7 Airlines, a Russian airline based around major cities in Siberia. It is critical to notice that in this visualization, node labels are not expressed next to nodes. On mouseover, the nodes show the airport's name, and the connected airports are also highlighted. We can see visual clutter in the overlapping edges, as can be expected from a dense network in an NL visualization. The interactive elements are necessary to make this visualization effective, a more in-depth discussion of interactivity in network visualizations found in section 4.3.2.

3.1.2 Social Network

Another typical example of a network represents a social network with edges between people who consider themselves friends. These are common for mapping out communities to understand relationships. While these are not used in a commercial sense, similar to the way the airline networks discussed in Section 3.1.1, they are incredibly useful in many academic contexts, from sociology to information visualization.

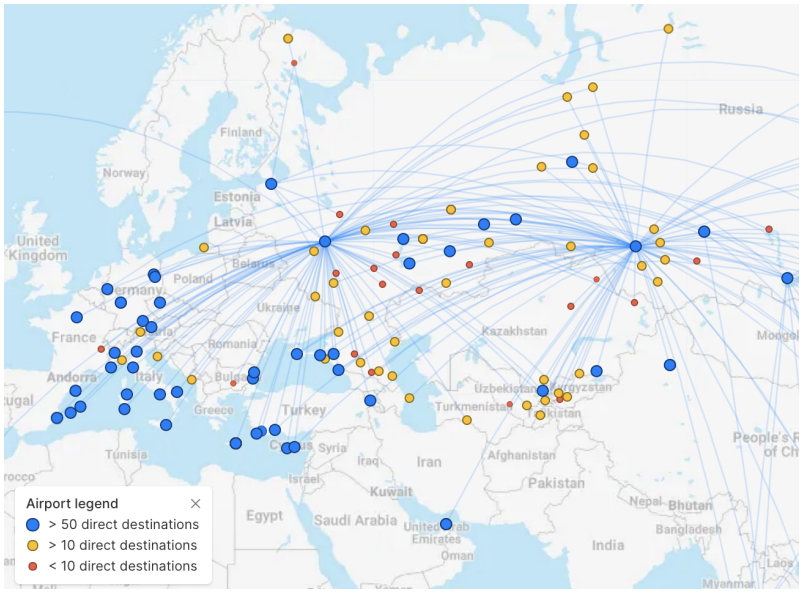


Figure 3.1: Example of airline network visualization of S7 Airlines [Con20].

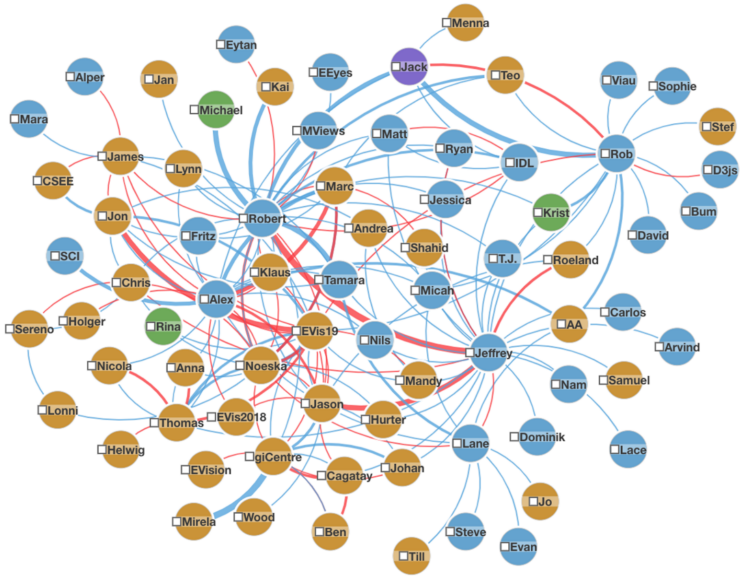
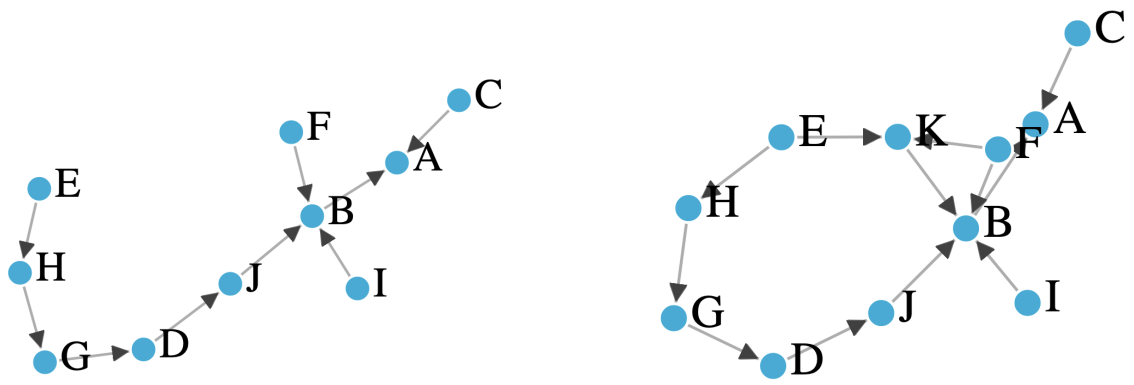


Figure 3.2: Example of social network visualization based on a collection of 30 twitter users [NWHL20].

Figure 3.2 contains a representation of a collection of twitter users. Each node is a separate twitter user, and the edges between the nodes represent retweets; the thicker the edge, the more retweets occur between those two users. Again, the NL visualization suffers from clutter, making it hard to distinguish minute details about the network. The Nobre *et al.* study on multivariate visualization efficacy used this network.

3.2 Node-Link

The node-link visualization of networks is fundamental; hardly any modern work has been written on the visualization's basic premise because of how fundamental the model is. At its core, it is a visual representation that draws nodes as circles and edges as lines; better put by Lloyd *et al.* in their textbook on graph theory, "a diagram consisting of a set of points together with lines joining certain pairs of these points" [LBM78]. That is not to say there has been no considerable study into node-link visualizations; while most graph theory works do not focus on the fundamentals of node-link visualizations, substantial research has been done into optimal ways of laying out node-link networks. For example, planar node-link visualizations focus on eliminating or, if that is impossible, reducing the number of edges that cross each other [NC88], as it is these cross edges that contribute to visual clutter and make the visualization challenging to follow. There has also been a considerable study into the differences between straight lines, curved lines, and cornered lines (that is, straight lines that are allowed to bend at 90-degree angles) in node-link visualizations [BETT94; XRP+12].



(a) Ten node example network with density 0.9

(b) Node K added to the previous nodes

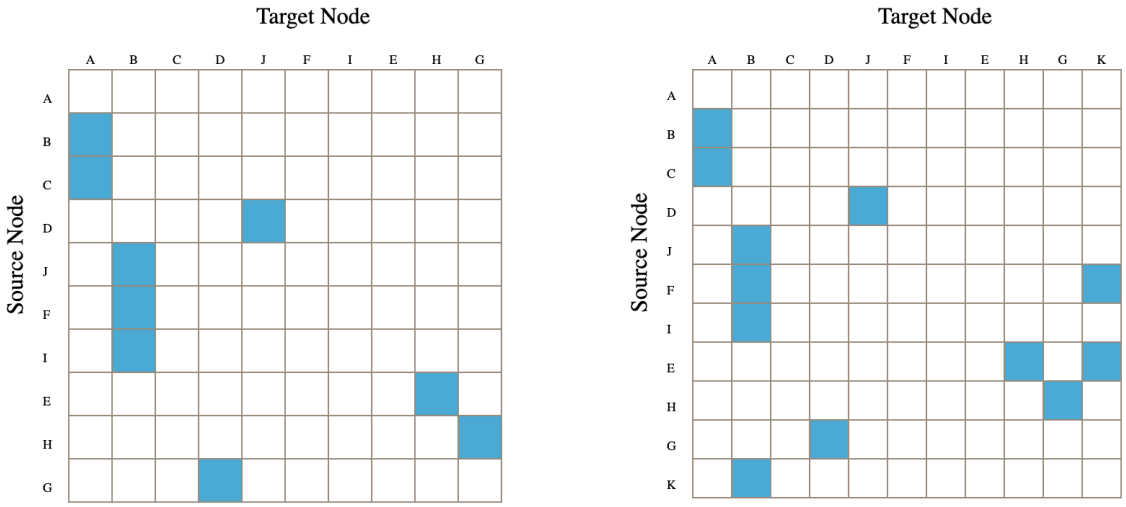
Figure 3.3: An example of adding a single node, K, with three edges to a NL visualization. The added edges are (K, B), (F, K), and (E, K). This network uses a force-directed layout.

Node link visualizations suffer from the issue that they do not scale well, with respect to visualizations that define expansion criteria. There is no clearly defined methodology to add an additional node on NL, except for a layout that specifically places nodes. For example, formatting all nodes equidistantly on a circle will easily define where to place an additional node; however, even in this example, adding a new node requires the movement of all other nodes in the circle. Alternatively, layouts such as the force-directed layout, which simulate forces applied between nodes, can redistribute the entire network, resulting in a network that appears to be entirely distinct, despite the only difference

being the addition of a single node [BGT13; Kob12]. NL visualizations also suffer from the issue of visual clutter. The more nodes or edges added to an NL diagram, the more cluttered and less readable the diagram as a whole becomes. Furthermore, adding additional edges, especially edges that overlap other edges increases this visual clutter further. Reducing visual clutter is a massive field of research in its own right [DSD16].

3.3 Adjacency Matrix

An adjacency matrix represents a graph, G, in an $n \times n$ matrix A, where n is the number of nodes in the graph. The ij^{th} entry, A_{ij}, defines if the edge (i, j) is in G [Wil96]. If a graph is undirected, then the adjacency matrix is symmetric, as there is no distinction between (i, j) and (j, i), hence $A_{ij} = A_{ji}$. In contract, directed graphs are asymmetric [NC88]. Typically, the row, i, represents a source node of an edge while column, j, represents the target node of an edge; in undirected graphs, there is no distinction between source and target nodes, hence the symmetric matrix. Adjacency matrices have an advantage over node-link, in that AM visualizations are suitable for dense networks. Due to how nodes are places in AM visualizations, we can add additional nodes without moving nodes already in the network. Additionally, we know exactly how the AM visualization will increase in size with the addition of new nodes. AM visualizations do not suffer from visual clutter in the same manner that NL visualizations do. Each edge has a unique defined position, and edges cannot overlap in an AM visualization. In this manner, AM visualizations are more visually efficient than NL visualizations [Tur84].



(a) Ten node example network with density 0.9

(b) Node K added to the previous nodes

Figure 3.4: An example of adding a single node, K, with three edges to a AM visualization. The added edges are (K, B), (F, K), and (E, K).

The most significant downside of AM visualizations is that they are not as intuitive as NL visualizations. In terms of gaze fixations, viewing two directly adjacent nodes and the edge connecting them in an NL visualization will require focusing on two nodes that are generally close to each other, with an edge directly connected to both nodes. The eye positions for looking at this

edge are all close together, effectively one massive gaze fixation. The AM visualization requires seeing an edge, then following the row to find the edge's source, and then following the column to find the target of the edge. These eye movements create three distinct gaze fixations, which is simply numerically more complex than the NL visualization. Finding a path between two nodes in NL requires following the path in a way that is intuitive to most people. In AM, it requires repeatedly switching between vertical and horizontal positions, which is not at all intuitive. Additionally, AM is not suitable for large graphs, as the size of the matrix grows quadratically with the addition of each node. In short, AM trades ease of understanding for compactness and conciseness.

3.4 Bipartite

The term "Bipartite" refers to two entirely different concepts within graph theory. In the interest of clarity, we address both concepts here; however, for this thesis's purposes, only the bipartite layout is used at large.

3.4.1 Two-Mode Networks (Bipartite Data)

Two-mode data, or bipartite data, refers to a network dataset that has two types of nodes. The term "mode" refers to a set of one type of data. Many major networks are one-mode networks, where all the nodes are of equivalent type. Two-mode networks will have two types of data [WF94]. For example, Ann *et al.* describes a flavor pairing network with two classes of nodes, one representing various foods and the other representing various flavors. Every food had edges to one or more flavors, but there were no edges between foods and no edges between flavors. In this example, one mode is the foods, and the other mode is the flavors, making it a two-mode dataset. This network can be seen in figure 3.5 [AABB11]. It is important to note that this does not end with two-mode datasets. If there are multiple different sets of data types, it is entirely possible to have a three-mode network (tripartite) or an n-mode network.

3.4.2 Bipartite Layout

The bipartite layout is a separate concept within graph theory, focusing entirely on a network's visualization. A bipartite layout can be applied to bipartite data or can be applied to regular one-mode data. Typically, a bipartite layout lists nodes vertically in two columns; this does not necessarily have to be arranged vertically, but this is the commonly accepted convention. With two-mode data, each column will represent nodes of a different mode, as shown in figure 3.5.

BP visualizations share many aspects with AM visualizations. They are generally more compact and concise with separate representations for nodes receiving and sending edges, while sharing some aspects with NL visualizations, such as having circles represent nodes and having edges that directly connect adjacent nodes. Additionally, nodes in both AM and BP have fixed positions making it easier to look for a single node with respect to looking for a single node on NL. Similar to the AM visualization, the BP visualization has a rigid and systematic methodology for adding new nodes, not requiring any other nodes' movement. BP visualizations additionally have a placement for node labels that do not potentially overlap with edges, meaning the issue of visual clutter is significantly

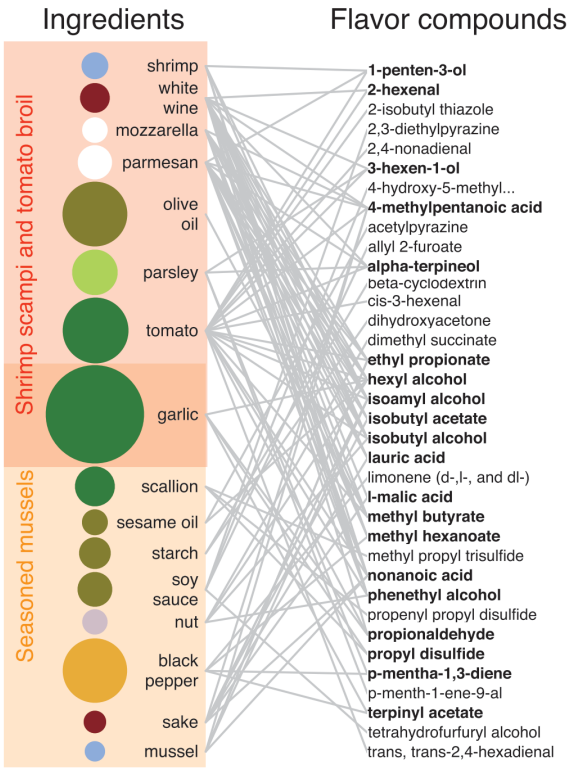


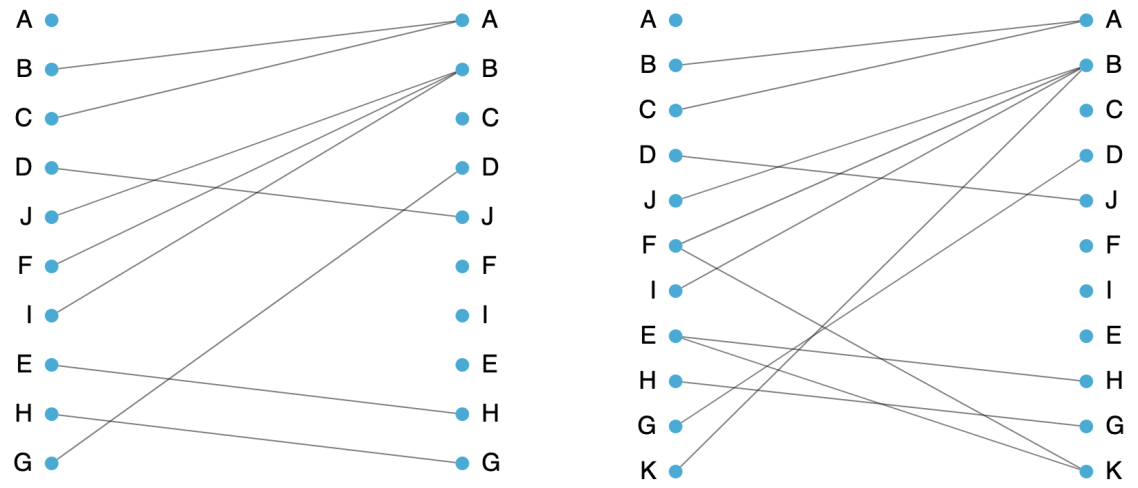
Figure 3.5: A two-mode network in a bipartite layout.

reduced on BP visualizations as compared to NL visualizations; however, because edges overlap in BP visualizations, visual clutter can still be a problem for very dense graphs. However, in contrast to AM visualizations, the BP layout is one dimensional; adding additional nodes only increases the vertical (in the above example) size of the network. As such, the size of BP layouts increases linearly with the size of the network, while AM visualizations increase quadratically with the size of the network.

3.5 Labeling

Labeling refers to writing the node names on the network visualization, not to the node names themselves. Node naming is a separate discussion that takes place in the following Section 4.2.5. Regardless of a node's name, whether it is a single letter, a number, a word, or a phrase, there is a decision to be made regarding whether this label should be placed in the visualization. In the case of NL visualizations, there is also a question of where the label should be placed, with the following common options [Hu09],

- Simple label placement



(a) Ten node example network with density 0.9

(b) Node K added to the previous nodes

Figure 3.6: An example of adding a single node, K, with three edges to a BP visualization. The added edges are (K, B), (F, K), and (E, K).

This method consists of placing the node in the same position for every node, for example, placing a label 5 pixels above the node. This method's advantage is that it is always possible to know which node a label refers to, as all labels follow the same placement rules. The main disadvantage is that the placement of labels can easily overlap with other nodes, labels, and edges resulting in visual clutter.

- Dynamic label placement

  This method consists of using nearby nodes, labels, and edges to allocate node label locations in such a way to minimize or eliminate overlapping. The advantage is that this reduces visual clutter and makes the visualization easier to understand. The disadvantage is that each label is in a unique position relative to its node, meaning that there can be confusion about which label corresponds to which node.

- Internal label placement

  This method expands the size of nodes to fit labels within the nodes' representations themselves. There are several considerations with this method. Namely, what size to make the nodes. If nodes are sized dynamically to be only large enough to fit the label, then nodes with longer names will have larger labels and thus be larger nodes. Typically, increasing the size of an individual node implies importance, meaning that nodes should be sized the same if this is not desired. However, this can cause all the nodes to be very large in size, decreasing space for edges, and ultimately increasing visual clutter.

There has been some study into whether or not labeling tasks improves performance. Jianu *et al.* tested whether node labeling improved memorability, as their study had tasks testing how memorable a network visualization was. They found that removing labels specifically causes a drop in memorability, a drop not found when other visual elements are removed. Additionally, they also did an ad-hoc analysis on whether labeling specifically improved accuracy or response

time. They found that there was some evidence to suggest that labeling makes little difference in non-memorability tasks; however, they did specify that they did not specifically focus on this and could not confirm the conclusion that labeling is not a major factor [JRHT14].

Node label placement is a massive field in its own right [Hu09]. This section hardly does that field justice with a light overview. The standard model for efficacy studies such as ours is to use the most straightforward label placement possible. There are already numerous variables and factors in efficacy studies; introducing label placement unnecessarily complicates the study.

Ultimately, for our study, we will use the simple label placement method. This strategy is a relatively standard practice for research in this space. Additionally, there are several arguably more important factors to consider outside of merely where to place labels. However, despite our using the most straightforward label placement scheme, this was not an arbitrary decision.

4 Related Work

This chapter covers previous NVE studies. We start with a broad overview of several previous studies we use to base our decisions. We then have an extensive discussion of the decisions made by previous studies with respect to various network properties, task selection, visualization properties and finally basic study design.

4.1 Overview

There has already been significant research into the efficacy of Adjacency Matrix and Node Link visualizations. The seminal work related to network visualization efficacy was by Ghoniem *et al.* [GFC04], where they evaluated both AM and NL visualizations. A total of nine different randomly generated undirected networks ranging in both the number of nodes (20, 50, and 100) and density (.8, 3.2, 7.2) were used against seven tasks to measuring both the accuracy and time needed to complete the task. The tests included basic interactivity elements, including being able to mouse over nodes to highlight them and their connections. Additionally, a link could be moused over to highlight both endpoints. Ghoniem specifically asked about forming a path between two nodes and about finding a common neighbor between two nodes. They found that AM universally performed better than NL for large, dense graphs, while NL outperforms AM in some times tasks on small, loosely connected graphs. NL outperformed AM on connectivity-related tasks [GFC04].

Similarly to this work, Keller *et al.* also examined the efficacy of AM and NL graph visualizations. Critical differences between Keller and Ghoniem include the use of directed, non-random graphs generated from real-world data and some differences in tasks. The two networks used were generated from a model of German roads consisting of 50 nodes with a density of 2.84 and a network based on the construction of a diesel engine, composed of 22 nodes with a density of 5.72. Critically for our purposes, tasks include asking about node in-degree, finding a common neighbor, and finding the shortest path between two nodes. The interactivity is nearly identical to that described by Ghoniem. Keller found that for finding the shortest path and common neighbors, NL consistently outperformed AM. This is consistent with the findings of other studies in this area. There was no statistically significant difference between AM or NL when looking at finding node in-degree [KEC06].

Expanding on the work of Ghoniem and Keller, Okoe *et al.* revisited these questions over a decade later. The major modifications on Ghoniem included expanding the sample size by using online crowdsourcing from Amazon's Mechanical Turk, adding additional questions, using a single significantly larger network based on real-world data, and adding new interactivity to the representations of the network. Okoe used an undirected network with 258 nodes and a density of 4.22 using our measurement; the network was built from food-flavor pairing data. Included in their tasks were cluster matching, finding common neighbors, and the shortest path between two nodes. Interactivity included the same interactivity as in Ghoniem, with an additional ability to

move nodes, zoom into and pan around the visualization field in the NL visualization. Okoe found that by and large, the NL graph statistically outperformed the AM graph on connectivity tasks. AM outperforms NL on cluster tasks [OJK18].

Ren *et al* . conducted a more recent study into network visualization efficacy. It looked into the efficacy of different AM node ordering, specifically ordering based on the hierarchical degree or clustering. They used two networks, both undirected but created from real-world data, one based on a 20 node network based on individuals in Kandahar province in Afghanistan with a density of .95, and the other a 50 node network created from an NSA training data set with a density of 1.42. The only interactivity included is a mouse over node feature, highlighting a node and all connected edges. The study consisted of 17 tasks, each related to a broad genre of a task, including critically for us, connectivity and clustering tasks. They found, again, that for clustering tasks, AM outperforms NL, and for connectivity tasks, NL outperforms AM; however, they found no discernible or statistical difference in how the AM representation was sorted. Additionally, the size of the network does not appear to create much of a difference [RMO+19]; however, this could be due to the difference in density, as was found by Ghoniem [GFC04].

Most recently, Nobre *et al.* studied network visualization techniques as they pertain to multivariate information. Specifically, they only focused on attribute specific tasks, such as finding specific attributes along the shortest path. They did not reexamine tasks similar to those looked at previously. Their network was undirected and generated from real-world twitter data, with 75 nodes and a density of 1.90. Critically, they introduced a series of new interactive features such as the ability to reorder the AM representation, an ability with no analog in the NL representation, and Nobre themselves points this out. However, despite the differences in tasks and interactivity, Nobre does find that for connectivity tasks (such as pathfinding and common neighbors), NL generally outperforms AM. For cluster-based tasks, AM outperforms NL [NWHL20].

4.2 Network Properties

This section focuses on the network properties found in various previous NVE studies. This discussion includes the data used to generate these networks, the size and densities of these networks, and other similar properties.

4.2.1 Network Data

Most studies in this field use real-world datasets, and there is some evidence to suggest that real-world datasets can improve "data attractiveness"; however, real-world datasets come with significant downsides. Real-world datasets often are far too large [APH17], leading to most studies using real-world datasets using smaller subsets of the data [BMK96; OJK18; RMO+19]. Real-world datasets also do not generally vary in size or density, making variability studies nearly impossible [APH17]. Possibilities for variability when using real-world data include using different subsets of the data to generate sub-networks with specific, desired size, density, and other properties [BMK96]. Another option is to find multiple datasets with different properties [KEC06; RMO+19]. Finally, real-world datasets can be incredibly domain-specific, requiring specific expertise even to understand the dataset. For example, the Keller study used a dataset surrounding the VistaD

Study	Ours	Blythe	Ghoniem	Keller	Okoe	Ren	Nobre
Size	20, 50	12	20, 50, 100	22, 50	258	20, 50	25, 75
Density	1, 2, 4	2	.8, 3.2, 7.2	5.72, 2.84	4.22	.95, 1.42	2.36, 1.9
Number of networks	6	5	9	2	1	2	2
Generation	FARZ	Real-world	Random	Real-world	Real-world	Real-world	Real-world
Density Measurement	$\frac{E}{V}$	$\frac{E}{V^2}$	$\sqrt{\frac{E}{V^2}}$	$\frac{E}{V^2}$	$\frac{E}{V^2}$	$\sqrt{\frac{E}{V^2}}$	$\sqrt{\frac{E}{V^2}}$
Directionality	Directed	Undirected	Undirected	Undirected	Undirected	Undirected	Undirected
Edge Weighting	X	X	X	X	X	X	✓

Table 4.1: Network properties of networks used in previous visualization efficacy studies [BMK96; GFC04; KEC06; NWHL20; OJK18; RMO+19].

model of a diesel engine, which can be incredibly complicated [KEC06]. Only participants with knowledge of the inner workings of an engine will gain any recognition benefit from the data. All other participants would only see the names as random; additionally, modifying these data, such as drawing a subset of nodes and edges to control the size and density, would require knowledge of the underlying dataset. Otherwise, a subset dataset can be created that would not make sense to someone with knowledge of the underlying data. Using the VistaD dataset as an example, should the subset network pulled from the overall network have missing nodes or edges that someone with knowledge of the overlying dataset could be left confused, which could have a deleterious effect on their response accuracy and time [APH17].

Generated datasets generally allow for absolute control over the properties of the data. However, the downside is the potential loss of data attractiveness as the information is entirely arbitrary. This could be offset by inventing meaning behind the dataset, however, there is some evidence to suggest that an invented meaning behind a generated dataset leads to a similar loss of data attractiveness [APH17]. Ghoniem used a generated network that was mostly completely random. They pulled their network from the ISPT Random Graph Server hosted by Waseda University [KN04]. This graph server was built based on Kawabe and Nimura's work, using the $G(n, M)$ random graph model. Critical additions were made by Kawabe and Nimura, in that they also added random edge weights between 0 and 1 to all edges with uniform probability [KN02]. Ghoniem did edit the dataset received from ISPT by adding an extra 10% of edges to the most connected node, adding these edges with uniform probability. In this way, Ghoniem created a pseudo-preferentially attached mostly random network [GFC04].

4.2.2 Graph Size

Starting with Blythe *et al.*, a study on the influence of node placement in a node-link network visualization, the study consisted of an overall network of 36 nodes. The actual study was performed on a subset of this overall network consisting of 12 nodes. The justification for this size was that it allowed them "hold structural relationships among nodes constant while varying their spatial relationships", and increasing node size would have largely limited this ability to vary spatial relationships significantly [BMK96]. Ghoniem used three different sizes: 20, 50, and 100 nodes. These sizes were specifically chosen to be able to test if the size was a factor in the data. They wanted

to ensure that network size did not introduce bias [GFC04]. Conversely, Okoe used a network with 258 nodes, compensating for any size issues with visualization and interactivity strategies. The justification for such a large network was that modern networks used outside of a study setting are generally vast and that limited testing on network visualization efficacy has been done on large networks [OJK18]. Keller had networks of size 22 and 50 while Ren used networks of size 20 and 50, with their reasoning being that they wanted to test networks of multiple sizes to compare the effects of network size on responses. [KEC06; RMO+19].

The most significant consideration with the size of the network comes down to "the capability of a visualization to handle an increasing amount of data", otherwise known as data scalability [LBI+12]. In theory, on an infinitely large display, an infinitely large data representation could be shown; however, given real-world limitations, dataset size needs to be managed. There are numerous visualization techniques to manage the amount of data shown in a limited screen size [YN06]. In network visualization, an incredibly common strategy is to limit the size of the network by reducing the number of nodes [APH17]. This can include choosing a smaller dataset [KEC06; RMO+19], or a subset of a dataset [BMK96; RMO+19]. The consensus across all works is that network size does increase the difficulty of a task, as evaluated through lower task accuracy and longer response times. However, the effect of increasing size is not as dramatic as the effect of increasing density.

4.2.3 Network Density

The density of a network describes the relationship between the number of nodes and the number of edges. There is no consensus as to how to calculate network density. We describe different density calculation methods in the following subsection. For the discussion here, we can convert all densities to use the density measurement that we use in this study, even if other NVE studies used a different network density calculation.

Ghoniem found that the network property that had the greatest effect on accuracy and response time was the network density; a small, dense graph was more difficult than a large, sparse one for most tasks. As such, the density of the test network is a major consideration. Ghoniem used density values of .8, 3.2, and 7.2 [GFC04]. Okoe's one network had a density of 4.22, a size decided by the data and not selected for nor controlled by Okoe [OJK18]. Ren and Keller had similarly sized networks; however, their densities varied wildly; Keller's networks had 5.72 and 2.84 for their small and large networks, respectively, while Ren had densities of .95 and 1.42 for their small and large networks. Notably, both Keller and Ren did not select networks based on density; they both found real-world networks and accepted the density those networks provided [KEC06; RMO+19]. This is one of the biggest downsides of using real-world data, the inability to control for all properties of a network [APH17].

Calculating Network Density

There appear to be three competing strategies for evaluating network density and considerable debate regarding the merits of each density evaluation technique. Traditionally, network density is described as a ratio of the number of edges in a network to the number of nodes:

$$d = \frac{m}{n} \tag{4.1}$$

where m is the number of edges, and n is the number of nodes. This method is incredibly simple to understand, as density is a simple measurement of the average number of edges per node; additionally, it is also a better description of density found in real-world networks. The primary fault of this method is that it provides no real sense of how "full" a network is. In general, most networks would be called "simple graphs", defined as a graph without loops or multiple edges between unique pairs of nodes. The only counter-example being directed networks, where the two edges (u, v) and (v, u) are distinct edges, solely because of the defined directionality. In general, simple graphs are undirected [Wil96]. However, it is fairly uncommon to see networks such that two edges, $(u, v)_1$ and $(u, v)_2$, between the same pair of nodes are considered distinct, separate edges, even in directed graphs. Because of this property, a network has a theoretical maximum amount of possible edges, as once an edge is drawn between every possible pair of distinct nodes, then no further edges can be drawn. This generally occurs at $|E| = \frac{|V|(|V|-1)}{2}$ for undirected networks [GS93], and $|E| = |V|(|V| - 1)$ for directed networks (this logically follows from the previous statement, as every edge in an undirected network can be drawn twice in a directed network, (u, v) and (v, u), hence there are double the number of possible edges). While this number exists and can be provided or generated by whoever is viewing a network, the actual density measurement offers no information specifically about how close the network is to maximum edge capacity [Mel06].

Another common method is to transform the density into a percentage:

$$d = \frac{m}{n^2} \tag{4.2}$$

This method also guarantees a value between 0, a network with no edges, and 1, a network with edges between every pair of nodes. The biggest advantage of this method is that it describes the density in terms of how full a network is. Every network has an upper limit on how many edges can be contained within that network, as each edge requires two nodes, and, at least in most graph theory contexts, multiple edges are not allowed between the same pairs of nodes. Granted, this same perception can also be found in the traditional ratio method. As d approaches n, the closer the network is to being filled. Blythe and Keller both use this methodology so that they can describe their networks in terms of percentages, giving an idea of how saturated the given network is with edges [BMK96; KEC06]. Okoe also uses this method, specifically discussing its relevance to real-world networks [OJK18]. The biggest downside of this method is largely the effect on how many edges there are to any given node. This information can be calculated from the provided density value but is not immediately apparent.

Being one of the first works in network visualization efficacy, Ghoniem's method has been fairly commonly adopted across many studies in this space. Their methodology is as follows:

$$d = \sqrt{\frac{m}{n^2}} \tag{4.3}$$

This method also guarantees a value between 0, a network with no edges, and 1, a network with all possible edges. Ghoniem stated that this methodology is topologically significant and scale variant, as the "number of potential edges increases in the square of the number of vertices" [GFC04]. Other works, such as Nobre's, have cited Ghoniem's position as an early work in this field and used this measurement. This is the only justification for using this measurement of density [NWHL20]. Ren also claims to use this measurement to compare to Ghoniem used it, but then performed a calculation that is entirely different and not described [RMO+19]. The justification for the square

root is that it limits the domain of numbers that can be used to describe density, thus causing the domain to be constant regardless of other network properties. It better describes the relationship between the number of nodes and edges, especially for large networks [Mel06].

We decided to use the classic ratio method for calculating density. We primarily used this method as we selected a generated network, and most generation algorithms use the classic ratio method as a network generation parameter. Additionally, we did not see the benefit of using a percentage method. We are interested in the effects of increasing density, and previous research has not shown that density as a function of the percentage of possible edges is a variable factor. That is to say, a network that is 50% full is not a point in which accuracy and response time decreases, merely that network density has an inverse relationship with task accuracy and response time. For these reasons, we believe that the classic ratio method of calculating density is the most effective method for our purposes.

Note: We will be using the classic ratio method for calculating density in this work; all density values have been adjusted to reflect how we measure density.

4.2.4 Network Directionality

As was mentioned in the discussion of simple graphs, the directionality of a graph is a boolean function stating whether or not edges are scalar values or vector values. That is to say, do edges contain directional information or purely connection information. Across all previous works, this property seems to have the most significant consensus across all network visualization efficacy studies. Ghoniem's network generation scheme was a random graph, a dataset that was randomly generated; there was no mechanism to add directionality to the edges [KN02]. Ghoniem could have dealt with a directional dataset by just randomly assigning the directionality of the edges. They specifically did not do this because it would have been unnecessary; none of the tasks related to directionality. While this does not preclude using a directed network, the two cited reasons for why an undirected network was used were that it simplified the node-link representation, as arrows were unnecessary. It simplified the shortest path task, as a direction did not need to be specified [GFC04]. Keller expanded on Ghoniem by examining direction relevant tasks and used a directed network [KEC06]. Some studies use an undirected network without offering reasoning as to why the network used was undirected. However, the reason can be perhaps inferred by using tasks that have no relevance to edge directionality [OJK18; RMO+19]. Nobre specifically used a directed dataset that was treated as undirected to simplify the network [NWHL20].

For previous studies, using directional data was a decision regarding visual clutter or clarity. The vast majority used undirected data or directed data where the directedness ignored to transform the network to an undirected network. However, unlike previous studies, we are using the BP visualization, and as discussed in 3.4, BP visualizations only really make sense with directed networks. Thus, in contrast to every other network visualization study, we will need a directed network. While BP visualizations can visualize undirected networks, there is a mirroring effect, similar to that found in AM visualizations of undirected networks 3.3. In early tests of our BP visualization, we found that visual clutter was a significant concern. Given that the edge mirroring caused by using an undirected dataset resulting in doubling the number of edges drawn in a BP visualization, we decided that it would be best to use a directed network.

4.2.5 Node Naming

Networks that were generated from real-world data generally used node names that corresponded with the data being modeled. For example, one of Keller's networks was generated from the VistaD model of a diesel engine; all of the node names are of various engine components, such as the "Sump", "Crankshaft", or "Flywheel" [KEC06]. Another example from Nobre includes social media data, where person names are used for overarching nodes, and the multimodal data uses self-descriptive names, such as "age" or "location" [NWHL20]. The justification for this type of labeling is not written explicitly; however, it can be seen why such names are included. It makes little sense to use entirely arbitrary naming schemes when relatable dataset names are available. However, to the best of our knowledge, there has not been study into whether this makes an actual difference in study results. There is a single counterexample to this rule, from Blythe; the study uses smaller networks pulled from an overall social network with named nodes. However, these smaller networks only pulled 12 nodes, with their names being mostly arbitrary. These smaller networks had their nodes mapped to a new name from a previous name, in an apparent attempt to protect privacy, though the exact reasoning is left unstated [BMK96]. This renaming strategy was similarly implemented by Ren, who related location names to generated fake place names, with the stated reasoning being that they wished to prevent any confounding effect from people recognizing a town's name [RMO+19].

Not all network visualization studies have used networks generated from real-world data and thus did not have access to already existing naming schemes. Ghoniem's random graph generation using the ISPT random graph server did not have naming implemented, instead of naming nodes after arbitrary numbers[KN04]. Ghoniem added labels were simple letters, capital 'A' through 'Z', for graph sizes 26 nodes and under; graphs over 26 nodes would match a letter and a number, so the fifty node graph used labels from "A1" to "F0" and one hundred nodes used labels from "A1" to "K0". There was no citation nor justification as to why this naming scheme specifically was used, nor is there any reasoning why "A0" was skipped [GFC04]. Keller claimed that "relatable, concrete dataset[s] might help users understand tasks better" [KEC06], sourcing this claim from a book written over presentations in a seminar on crowdsourcing and human-centered experiments. The book written by Archambault *et al.*, included a section on real-world datasets, offering that real-world datasets increase data attractiveness, which is a key feature of data necessary to improve motivation for crowdsourced studies; however, they also included that real-world datasets have many drawbacks, among which naming is one. Names may not be relevant to participants, meaning those names lose all relevance to participants, and those names might as well effectively be random. Additionally, several other options for improving data attractiveness are provided, which have to do with study design and not labeling. Overall, random labeling appears not to be a confounding factor in information visualization [APH17; JRHT14].

4.2.6 Edge Weighting and Other Properties

Another property worth mentioning is edge weighting. This generally makes more sense with real-world data, as a connection cost or benefit between two nodes. For example, a network representing a toll road network may have edge weights representing the cost of the toll roads. However, most studies on network visualization efficacy do not include simple edge weights [GFC04; KEC06; OJK18; RMO+19]. Nobre was specifically looking at including multivariate

data within a network, which is theoretically similar to edge weighting while being significantly more elaborate. The data provided were not simple single values, but instead, a series of values, presented with different methods. A key aspect of the Nobre study was the different representations of this multivariate data. As such, there is little analog between this an edge weighting [NWHL20]. As of yet, the only study with even a remote comparison to edge weighting was Nobre. Other than that, all other visualization efficacy studies have included unweighted edges. We saw no benefit to including weighted edges in our study, especially considering that we did not intend to pose an edge weight related task.

As was stated in the introduction of the previous section, we have only included properties that are directly relevant to NVE studies. This is not to say that there are no other properties of networks worth discussing. For example, we have only discussed the properties of static graphs; dynamic graphs are those that change over time, creating a whole host of new properties to describe them [BBDW14]. We have chosen not to include these properties as they are not directly applicable to a discussion on NVE at this time. Studies into network visualization efficacy have not fully assessed the effects of the properties described above on efficacy. Given that our contribution focuses on a different visualization and the general effects of interactivity, we will not highlight the properties of graphs that other NVE studies do not highlight. Future work into NVE may include a discussion and analysis of these properties, and that discussion can be found in Chapter 7.

4.3 Color and Interactivity

This section focuses on the color and interactive elements of the network visualizations of previous NVE studies.

4.3.1 Color

Color is another massive consideration in network visualization. First and foremost, whether or not to include color at all is a decision to be made. If color is to be included, the chosen colors need to be such that colorblind individuals are not disadvantaged. Given the many different types of colorblindness, this color-based inclusivity can be a real challenge in visualization design. Just as with the previous section, studying the effect of color on network visualization is a field in its own right [BEW95].

Continuing this discussion within the context of NVE studies, there has been no study into the effects of color on network visualization efficacy. None of the cited examples thus far have included any research into the effects of color. Almost all of these examples have some elements of color in their studies; for example, Ghoniem has mouse highlighted nodes outline in green while Okoe used different colors to establish different groups [GFC04; OJK18]. Jianu had an in-depth study examining the efficacy of different node-link visualization strategies, but did not include a group without color; that is, the four visualizations studies all included the same colors, just presented in different ways. The only major consideration Jianu presents is being mindful of colorblindness [JRHT14]. Archambault expands on this, saying that, especially for crowdsourced studies, color blindness can be a factor in participants, not to be over-reliant on color to distinguish elements. Using two different colors is nearly always acceptable, as most colorblind people can distinguish

between two differently colored elements, though this does depend on the differences between the colors. They also suggested having some other elements besides color to distinguish elements, such as labels or shapes [APH17]. Overall, including color has not been a serious consideration for any such study in this space.

4.3.2 Interactivity

Interactivity is one of the largest points of consideration within the visual elements of a network representation. Given that interactivity can make tasks significantly more comfortable to complete, it is critical to ensure that the provided interactivity does not become a confounding variable. There has already been significant study into interactivity as it pertains to network usability, with a consensus showing that interactive elements tend to ease network understanding so long as the interactive elements themselves are not difficult to understand [DS13].

	Ours	Blythe	Ghoniem	Keller	Okoe	Ren	Nobre
Node mouseover	X	X	✓	✓	✓	✓	✓
Edge mouseover	X	X	✓	✓	✓	✓	✓
Zoom into network	X	X	X	X	✓	X	✓
Node dragging	X	X	X	X	✓	X	✓
Reorder/Sort	X	X	X	X	X	X	✓

Table 4.2: Interactivity used in previous NVE studies [BMK96; GFC04; KEC06; NWHL20; OJK18; RMO+19].

Shifting this discussion to focus specifically on previous NVE studies, Ghoniem's interactivity was fairly basic, with nodes and edges highlighted when moused over and basic node selection and deselection by clicking; the stated reasoning for this interactivity was that participants were losing focus in a preliminary study [GFC04]. Keller included the same node selection and deselection but did not include any other interactive elements [KEC06]. Okoe's interactive elements were fairly significant, justifying having interactive elements as previous studies by Keller and Ghoniem had both used interactive elements; however, Okoe used several different commercial graph visualization tools and took the common interactive elements from these tools. The justification was that modern use of networks would generally include these elements, so when looking at efficacy, it follows that visualization efficacy should include common visualization elements. The elements used include hovering over nodes to highlight connected edges, node selection and deselection, and in the case of the node-link representation, the ability to zoom into the network and move nodes around [OJK18]. Ren also has the fundamental interactions found in the studies of Ghoniem and Keller [RMO+19].

Nobre had by far the most significant amount of interactive elements, with a broad ability to interact with the network. Nobre included all the previous interactivity. Also, the ability to select multiple nodes, select entire groups, and, in the case of adjacency matrices, reorder the rows and columns. Their paper stated that the ability to reorder columns according to connected nodes is an ability not

paralleled in the node-link representation, but they concluded that this was not a critical factor and chose to continue with their interactive elements as is. Their interactive elements were so complex that they required a 15-minute training session before the actual study to train participants on using the visualizations. This training is not unique to Nobre as most other studies have some training involved; however, Nobre was unique in how extensive the training was [NWHL20]. The issue with such extensive interactivity is that the results may not speak to the visualizations themselves, but instead the interactive tool. While this does not invalidate the results, it does call into question what the study as a whole was evaluating, the visualizations or the interactivity.

Some study has been done into the effect of interactivity on response time and accuracy. A study by Baranauskas looked at a static adjacency matrix representation compared with an interactive one. They found that the interactive matrix representation outperformed the static representation for some tasks, while there were no tasks in which the static representation outperformed the interactive representation. There were limits to this interactivity, including only the base elements of being able to mouse over nodes to highlight edges, select nodes, and move nodes [BPA+07]. This study again calls into question Nobre, who went beyond the norm of interactive elements. In a blog post after the Nobre study was published, Alexander Lex, one of the study's co-authors, argued that interaction design could be far too complex to isolate as a factor in the efficacy of a network visualization. Additionally, stating that the original study followed precedent by basing their interaction design on previous examples, though this does not appear to be the case [Lex20].

4.3.3 Other visual elements

The list of visual element considerations as it pertains to network visualization is endless. We briefly mentioned one in Section 3.5, with what size to make nodes. Edge thickness, overall graph canvas size, edge-node connections, cluster visualization, and more are all examples of network visual element considerations. We have only mentioned a few critical examples as they pertain to the research conducted here. It is important to be mindful that these considerations were non-exhaustive, and that we carefully considered every decision regarding the visual elements of our network visualization.

4.4 Task Selection

In this section, we address the decisions behind the tasks selected by previous NVE studies.

4.4.1 Number of Tasks

A key metric in the tasks of network visualization efficacy studies is quite simply the number of tasks used to evaluate network visualizations. Few tasks could indicate a lack of depth in testing the visualizations, leading to results that are somehow skewed. Many tasks could confound effects with study participants having an issue with a lack of focus, or just getting bored [APH17]. It is important to note that the number of tasks is not the only metric when evaluating tasks in these studies; only one of several key metrics. Ghoniem had seven tasks, each task focusing on separate pieces of information that could be taken from a network visualization; there was little overlap

between the tasks. Part of the reasoning behind a lack of overlap between tasks was that Ghoniem had seven tasks on nine different graphs resulting in a total of different tasks of 63 across each visualization [GFC04]. Keller had similar reasoning with only 6, but across multiple different networks [KEC06]. Okoe only had a single network and a desire to confirm the results of Ghoniem and explore precisely where the differences between different visualizations fall.

Okoe's tasks had overlapping requirements, for example, having one task requiring the participant to find all neighboring nodes of a highlighted node and another requiring finding all neighboring nodes of two highlighted nodes. These overlaps are meant to examine if slight differences in tasks will find similar differences in accuracy or response time. Okoe had 14 tasks, but only a single network, with a total number of tasks per visualization of 14 tasks, significantly less than Ghoniem [OJK18]. Likewise, Ren had 16 tasks across two different networks, implementing a similar strategy to Okoe of looking for differences by subtlety changing how questions were asked. Ren more explicitly organized tasks into overall categories. Ren did have two networks, with a resulting total tasks per visualization of 32 [RMO+19]. Nobre similarly had 16 tasks; however, given the goals of the Nobre study were primarily examining the more complex multivariate data, a majority of the tasks were about the multivariate data contained within the network itself, not about general network properties; for example, asking about the age of a particular person in the network, not just asking if a particular person was in the network [NWHL20]. It appears that, in general, more tasks are used when there are fewer networks to test on, as the differences between tasks are used to probe the different efficacies of the visualizations. In contrast, when multiple networks are available, fewer tasks can be used as both the differences between the tasks and the differences between the networks can be used to probe the efficacies of the visualizations.

4.4.2 Task Type

In these studies, tasks can be broadly organized into generalized types depending on what the task asks the participant to do or what skill the task requires to be completed. As was examined in the previous section, in general, when more networks are available, there tend to be fewer tasks per type, while when fewer networks are available, more varied tasks per type are used instead of separate networks. There are several ways to separate tasks into different types, and there is no consensus on how this separation should be conducted. For example, Ren separates tasks into many categories that would have all been labeled at "taxonomy" by Okoe [OJK18; RMO+19]. Several studies did not classify tasks at all [GFC04; KEC06]. Overall, the task separations applied here are meant to further discussion and are not meant as a suggestion of a task classification system to be applied more broadly.

Property

Tasks examining network properties are some of the simplest tasks. Examples include tasks asking about the basic properties of the network, such as the number of nodes or the number of edges in the network as a whole, both tasks used by Ghoniem. The stated result for using these tasks was to evaluate whether basic properties could be quickly found given different visualization methods. Another potential example of this is finding a specific node, another task used by Ghoniem and also by Keller [GFC04]. While finding a specific node is not a general network property, the task itself is more similar to those of finding a general network property than it is any other type of task; this

differs from a task to find a specific edge, which would generally fall under the "Connectivity" task type. Okoe did not include a single task that would fall under this task type [OJK18]. Ren asked four questions that would be generally classified under this type of [RMO+19]. We include a single task asking about the incoming links a highlighted node has.

Connectivity

Given the primary goal of networks, and arguably the most critical aspect of networks is to present the connections between nodes, it follows that tasks surrounding the connectivity of these nodes would be of primary importance to network visualization efficacy studies. Tasks asking about finding edges, finding neighbors or common neighbors, and finding a path between nodes would generally be considered connectivity tasks. In this vein, 14 of Ren's 16 tasks would be broadly considered connectivity tasks, though Ren sub-classifies these tasks differently into five different categories, based on the skills needed to accomplish these tasks [RMO+19]. Ghoniem's remaining 4 of 7 tasks were all connectivity tasks, and roughly half of Okoe's tasks were connectivity based [GFC04; OJK18]. We included two connectivity tasks asking to find a common neighbor between two highlighted nodes and finding the shortest path between two nodes.

Cluster

Cluster-based tasks are tasks regarding clusters of nodes that exist within the larger network. These tasks might include asking about which cluster a node is present in or how many clusters a network has; the latter of these tasks would not be considered a network property task unless the presented network somehow explicitly distinguished the clusters, such as the methodology behind Okoe's visualizations. Okoe is thus far the only study to explicitly looked at clustering in networks, asking questions such as if two highlighted nodes were in the same cluster [OJK18]. We included a single clustering task, simply asking if two highlighted nodes were in the same cluster.

Memorability

Memorability tasks include any task in which the participant must recall information after some time. This could include recalling information later in the study or recalling information in a follow-up after the study has concluded. Okoe explicitly had a single memorability question [OJK18]. Ren had a post-survey questionnaire where memorability was tested [RMO+19]. Keller and Ghoniem did not include any questions on memorability [GFC04; KEC06]. Overall, if memorability is included, it seems to be included in the same study, with only a gap of a few minutes between presenting the information and testing. This could be due to the difficulty of conducting follow-ups on crowdsourcing, or that follow-up information may be unreliable when crowdsourced [APH17].

We did not include any memorability tasks. There have been few NVE studies comparing AM and NL visualizations that ask about memorability. Given that our primary goal is to understand the efficacy of BP in various tasks specifically, a lack of previous data comparing AM and NL visualizations will largely prevent us from being able to draw conclusions regarding the efficacy of BP visualizations. Additionally, one limitation of using a crowdsourced sample is the difficulty of following up with participants, compared with in-person studies where it is possible to track down

participants or participate in a follow-up study a term of participation. In crowdsourced studies, it may be impossible to get participants to take part in the follow-up study, or even if participants do complete the follow-up, there can be significant differences in the times when the follow-up is completed, meaning the data received from the follow-up is limited in value[APH17]. Overall, because of the limitations we are faced with, we believe that the best decision is to focus our efforts on alternate tasks.

4.5 Study Method

These network visualization efficacy studies were structurally very similar, but one key difference between them was the test samples used in the survey. These differences informed many critical decisions made in our study's design or even what the results ultimately show. The samples are divided into two groups, studies conducted in a laboratory and crowdsourced studies.

4.5.1 Lab controlled studies

Of all the studies discussed above, two by Ghoniem and Keller conducted in-person studies. The Ghoniem study was conducted on 36 people who were all post-graduates or researchers in computer science. Every member of the population sample had previous exposure to graph theory [GFC04]. Similarly, the Keller study had two separate populations for each survey, with populations of 21 and 16. Each of these studies was completed by engineering Ph.D. students or professionals, with roughly a third of each group familiar with graph theory [KEC06]. Overall, these studies were mainly conducted before modern crowdsourcing tools were created, meaning their options for finding a sample population was more limited than modern researchers might face. However, conducting in-person allows for considerable control over the sample population, something mostly impossible to do in crowdsourced studies [APH17].

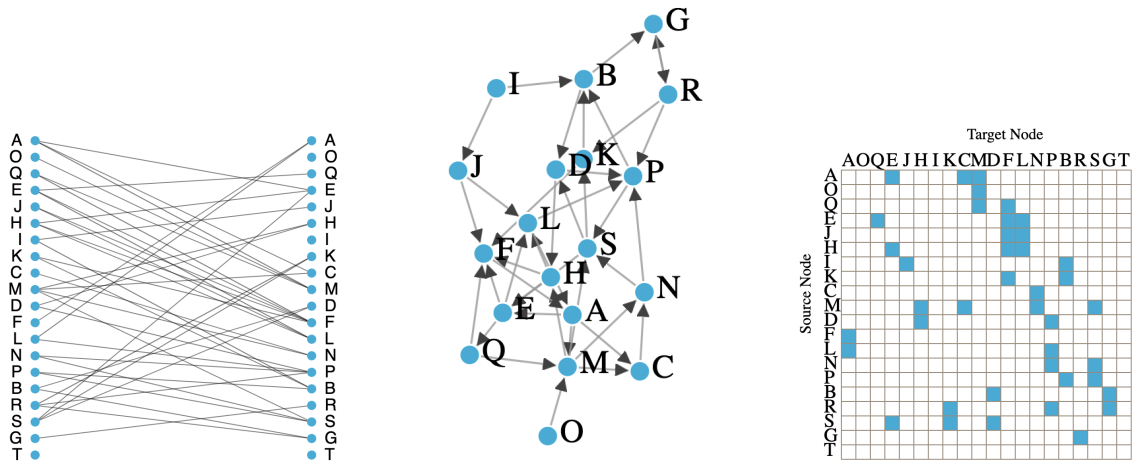
4.5.2 Crowdsourced Studies

The remaining studies discussed in this section have all been crowdsourced, all through Amazon's Mechanical Turk. Nobre had the smallest sample size at 322, while Okoe and Ren both had sample sizes of 557 and 600, respectively. All three studies required the authors to invalidate several responses, an issue mostly not faced by traditional in-person studies. Additionally, all three were unable to control for previous knowledge or even able to largely screen participants other than by location. Mechanical Turk does provide the ability for workers to self-report qualifications such as degrees. However, there is no verification method for this; it is an honor system [Cro12]. All three studies had large sample sizes to account for confounding effects in a crowdsourced sample [NWHL20; OJK18; RMO+19]. All crowdsourced studies face similar challenges in screening participants [APH17].

5 Network Data

There are numerous methodologies for generating graphs. Depending on the final resultant network's desired properties, many different network generation schemes can be used. Additionally, most of these schemes have mechanisms to further shape and refine the resulting network. The most significant advantage of generated networks is that they allow exact control over the data; whatever properties are desired in the test dataset can be generated by some schema. The most notable disadvantage is that the generated data holds no meaning [APH17], which may result in worsened test accuracy and response time; however, there is evidence suggesting that this is not the case [JRHT14].

5.1 Erdos-Renyi



(a) Bipartite visualization of Erdos-Renyi model

(b) Node-link visualization of Erdos-Renyi model

(c) Adjacency Matrix visualization of Erdos-Renyi model

Figure 5.1: The Erdos-Renyi Model on all three visualizations with a network size of 20 and a density of 2

The most common Random network generation scheme is probably the Erdos-Renyi model. The shown example in Algorithm 5.1 is for a directed Erdos-Renyi network, as described by Erdos and Renyi in their 1959 paper "On random graphs". The model works by checking all possible edges and adding an edge with probably p. If p is 1, then every possible edge that can be added will be added. If p is 0, then the network has no edges. This model limits control over the total number of edges, instead leaving the generated network to an expected value of edges; additionally, this network also leaves less control over the generation mechanics of individual edges [ER59].

Algorithm 5.1 Erdos-Renyi Network Generation [ER59]

```
procedure GENERATE ERDOS-RENYI NETWORK(n, p)
    G ← Graph()
    for i ∈ [1, ..., n] do
        G.addNode(i)
    end for
    for i ∈ [1, ..., n] do
        for j ∈ [1, ..., n] do
            if i ≠ j ∧ random < p then
                G.addEdge(i, j)
            end if
        end for
    end for
end procedure
```

Ultimately, we decided against the Erdos-Renyi due to the lack of control we can exercise over the network's parameters. There is no way to ensure clustering, and the completely random nature by which the network is created can result in networks with some strange properties, which may ultimately have a confounding influence on the data.

5.2 Watts-Strogatz

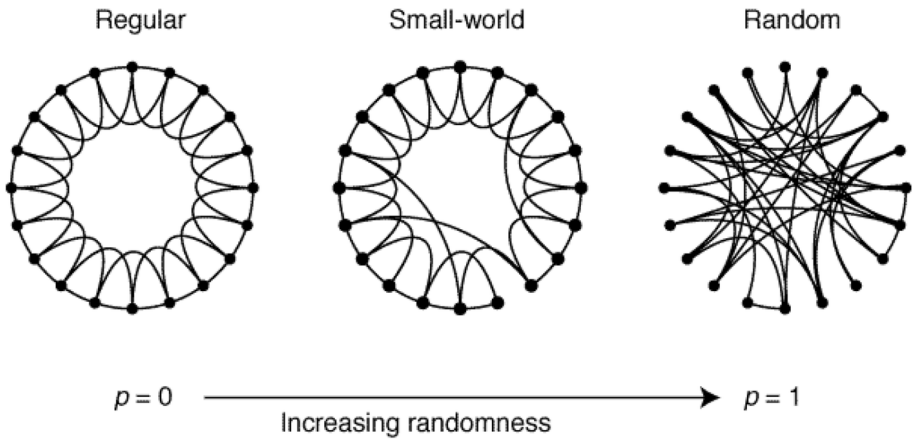


Figure 5.2: Model of Watts-Strogatz Network generation [WS98].

The advantage of the Watts-Strogatz network is that the network generation model is specifically designed to generate small-world networks, in which most people are connected to the nodes "nearest" to them. This model works by adding m edges to every node. With p probability, edges are added to either the nearest node or to a random node in the network. If p is 0, then the network is perfectly regular, with every node connected only to the m nodes closest to it. If p is 1, then the network is completely random. Even if p is 1, the random Watts-Strogatz network is unlike Erdos-Renyi, as Watts-Strogatz offers the ability to define the total number of edges, thus offering

direct control over the density of the network [WS98]. The biggest issue with Watts-Strogatz networks is that the network generated using that model does not reflect real-world networks. Watts-Strogatz does not create clusters, and given our desire to have a cluster related question, as will be discussed in Section 6.1, this network will not meet our needs.

Algorithm 5.2 Watts-Strogatz Network Generation [WS98]

```
procedure GENERATE WATTS-STROGATZ NETWORK(n, m, p)
    G ← Graph()
    for i ∈ [1, ..., n] do
        G.addNode(i)
    end for
    for i ∈ [1, ..., n] do
        for j ∈ [1, ..., m] do
            if random < p then
                G.addEdge(i, Nearest neighbor)
            else
                G.addEdge(i, Random node in G)
            end if
        end for
    end for
end procedure
```

5.3 Barabasi-Albert

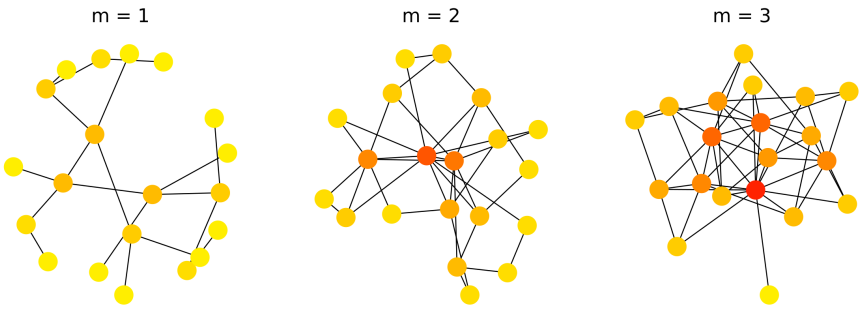


Figure 5.3: Model of Barabasi-Albert Network generation with increasing values for m. We can see the few highly connected nodes in red while the least connected nodes are yellow [Wik17].

The Barabasi-Albert model is primarily built around preferential attachment; that is to say, a node with many edges already connected to it will be more likely to have further edges connected to it. This preferential attachment should create a network that is scale-free, which means a network with a few nodes that connect to many nodes; another way to think about this is to think of a hub

and spoke, where a "hub" node connects to many different "spoke" nodes and the network as a whole has a few "hubs" with many "spokes". This model works by adding nodes one at a time and adding m edges to the previously added nodes. These target nodes are selected preferentially, where higher degree nodes are more likely to have an edge added to them [AB02]. At high density, the Barabasi model can exhibit some clustering behavior, but it is not entirely clear which nodes are in which clusters; especially at low density, it was impossible to discern which nodes were in which clusters. If we could not empirically determine the cluster, it would be impossible to assess if participants can correctly determine clusters. As such, this network generation model would not suit our purposes.

Algorithm 5.3 Barabasi-Albert Network Generation

```
procedure GENERATE BARABASI-ALBERT NETWORK(n, m)
    G ← Graph()
    for i ∈ [1, ..., m] do
        G.addNode(i)
    end for
    for i ∈ [m + 1, ..., n] do
        CONNECT(G, m)
    end for
end procedure

procedure CONNECT(G, m)
    L ← []                                          // Initialize choosing list
    for i ∈ G.nodes do
        L.add(i)          // Every node starts with equal probability of being chosen
        for (j, i) ∈ G.edges do
            L.add(i)                                // Weight nodes with more incoming nodes
        end for
    end for
    for i ∈ [1, ..., m] do
        g ←GETRANDOMITEM(L)                         // Choose a random node from the choosing list
        G.addNode(g)
        L.remove(g)                                 // Remove chosen node from choosing list
    end for
end procedure
```

5.4 FARZ

The FARZ model, named after the authors of the paper proposing the model, creates networks around having cluster groups. The algorithm takes three inputs, n, the number of nodes the network should have, m, the number of edges per node, and k, the number of clusters that the network should contain. These features allow for the creation of networks with specified densities, the m argument. This network also has an internal argument of β, allowing for precise control of the degree of separation between different clusters [FARZ18]. The advantage of this model is the built-in clustering mechanism. Instead of either defining clusters after a network is generated

Figure 5.4: Model of FARZ Network generation, showing an increasing clustering coefficient controlled by changing internal generation values [FARZ18].

or building a clustering mechanism into another network generation scheme, this model is built around clustering. Given that one of our tasks concerns clustering, we see it as necessary to select a network generation method that will create clusters that can be identified. We have written the FARZ Algorithm in detail in Algorithm 5.4; we slightly modified the `assign` procedure to be a simple random selection instead of a more complex selection model described by Fagnan *et al.*; otherwise, this is the exact algorithm described by Fagnan [FARZ18].

Algorithm 5.4 FARZ Network Generation

procedure GENERATE FARZ NETWORK(n, m, k)
  $G \leftarrow$ Graph()
  $C \leftarrow \{c_1 = \emptyset, c_2 = \emptyset, \ldots, c_k = \emptyset\}$
  for $i \in [1 \ldots n]$ **do**
    G.addNode(i) // Adds node i
    ASSIGN(i, C)
    CONNECT(i, C, G)
    for $[2 \ldots m]$ **do**
      $j \leftarrow$ select$G$.nodes
      connect(j, C, G)
    end for
  end for
  return G, C
end procedure

procedure ASSIGN(i, C)
  $L \leftarrow []$ // Initialize choosing list
  for $c \in C$ **do**
    L.add(c) // Every community starts with equal probability of being chosen
    for $j \in c$ **do**
      L.add(c) // Weight communities with more nodes
    end for
  end for
  $c \leftarrow$GETRANDOMITEM(L) // Choose a random community from the choosing list
  // This is implemented as a simple random selection
  c.add(i) // Add i to selected community
end procedure

procedure CONNECT(i, C, G)
  $\beta \leftarrow i \in [0, 1]$ // β determines how disjointed or connected communities are
  if random $< \beta$ **then** // Choose a community from,
    $c \leftarrow$ select($\{c, \forall c \in C \wedge i \in c\}$) // memberships of i
  else
    $c \leftarrow$ select($\{c, \forall c \in C \wedge i \in c\}$) // other communities
  end if
  $j \leftarrow$ choose($\{j, \forall j \in c \wedge j \neq i \wedge (i, j) \notin G$.edges$\}$) // Choose a node from the chosen community
  G.addEdge(i, j)
end procedure

6 Study Design

In this chapter, we describe the design of the study we performed. We cover all the decisions we made regarding the design of the efficacy study along with our reasoning behind those decision. We also extensively describe the exact procedure we followed when conducting the study.

6.1 Tasks

We decided on four separate tasks. We selected our tasks to cover multiple task types and to be able to compare the different visualizations. Given that we have six different networks, we have a total number of tasks per visualization of 24. To prevent a memory effect, we have participants only perform tasks on a single visualization, making a total number of tasks per participant of 24. Given that we specifically did not want to include interactivity, and Ghoniem stated that having a significant number of tasks without interactivity leads to a high level of frustration [GFC04], we decided that having a few carefully selected tasks would result in better quality data.

Tasks we included	Ghoniem	Keller	Okoe	Ren	Nobre
Shortest Path	✓[a]	✓	✓[a]	✓[a]	✓[d]
Incoming Links	✓[a]	✓[b]	✓[a,c]	✓[a]	✓[d]
Common Neighbor	✓[a]	✓	✓[a]	✓[a]	✓[d]
Same Group	X	X	✓	✓	✓[d]

Table 6.1: A comparison of our tasks to other NVE studies [GFC04; KEC06; NWHL20; OJK18; RMO+19]. (a) Task was for undirected graph, subtle differences in task requirements. (b) Task was for both incoming and outgoing links combined, not just incoming links. (c) Given two nodes, select the one with higher degree, not find the specific degree of one node. (d) All tasks were through the lens of other attributes contained within the network.

6.1.1 Shortest Path (SP)

This task asks participants to find the shortest path between two highlighted nodes. In our version of this task, we provide the direction that the path must take by specifying the starting and ending nodes. NL visualizations require participants to follow the arrows between two nodes. This is only complicated on large, dense graphs as visual clutter of multiple overlapping links and nodes can cause participants to be unable to follow a path. AM visualizations require participants to find

outgoing edges from the source or incoming edges from the target, and then find the corresponding column to that link to find the corresponding row and repeat the process. Overall, this switching between vertical and horizontal axes is unintuitive and complex. BP visualizations are fairly similar to AM visualizations, with the one change of BP visualizations only having a single axis.

6.1.2 Incoming Links (IL)

This task asks participants to find the in-degree of a specific node. Given a highlighted node on one of the visualizations, the participant needs to count the number of incoming links and then provide that number as a response. NL visualizations require participants to differentiate between incoming and outgoing links, a distinction made with an arrow. Very dense NL visualizations have difficulty due to the visual clutter caused by multiple overlapping links. AM visualizations require participants to find the appropriate column and then count all the edges in that column. BP visualizations require participants to find the appropriate node in the "incoming" column, and then count the attached edges, which can be difficult on large dense graphs.

6.1.3 Common Neighbors (CN)

This task asks participants to find the common neighbors of two highlighted nodes. Common neighbors are defined as: a node with two outgoing edges that connect to both highlighted nodes, or a node with two incoming edges coming from both highlighted nodes. We specifically do not include a node with one incoming edge and one outgoing edge from/to the two highlighted nodes. NL visualizations require the participant to follow all incoming and outgoing edges from each of the two highlighted nodes; however, NL visualizations also have a slight advantage as our NL visualization is laid out with a force-directed layout, meaning that common neighbors are more likely to be closer to the two highlighted nodes. This is not the case on the other two visualizations. AM visualizations require participants to follow both rows and columns for each highlighted nodes and see if both rows/columns have an edge to the same node. This is arguably the easiest technically to complete, as there are no visual clutter issues, and the process is clearly defined; however, this process is also entirely unintuitive and would not be apparent to any participant without previous knowledge of AM visualizations. BP visualizations require participants to perform a similar process to NL visualizations; however, BP visualizations are far more susceptible to visual clutter on dense networks, as it becomes nearly impossible to follow edges that go through a massive cluster of overlapping edges.

6.1.4 Same Group (SG)

This task asks participants whether two highlighted nodes are in the same group. Group clusters are defined at network generation as a function of how the network is generated. See section 5.4 for details about the network generation algorithm. We ask participants to use their best judgment in making this determination, and clusters are groups of nodes with many edges between them. We additionally advise that nodes in the same group tend to be closer together in the visualization. There are several different methodologies for figuring out if nodes are in the same cluster. On sparse graphs, this task can be largely impossible to determine, as there is not enough information

to decipher which nodes are in which clusters. On dense graphs, it is often significantly easier to see where the cluster boundaries lie. On NL, clusters will be shown as groups of nodes with several overlapping links and fewer links between groups. On AM, clusters will form boxes in the visualization. On BP, clusters will have separated boundaries of few links that can be seen in the web of links between the two columns of nodes. On AM and BP visualizations, we use hierarchical clustering to order the nodes, which almost universally results in nodes of the same cluster being positioned close together.

6.2 Hypotheses

Given the results found in previous studies, we expect that NL visualizations should generally outperform the AM visualizations at the shortest path task and common neighbor task, the two connectivity tasks. AM should outperform NL at the incoming links task, the same group task, the property, and cluster tasks. We also expect that as density and size increase, the performance of AM should not decline as rapidly as the performance of NL. These expectations are formed from the results of several previous studies [GFC04; KEC06; OJK18; RMO+19].

The specific hypotheses we have regarding BP visualizations are as follows,

1. On the Shortest Path (SP) task, BP will perform better than AM but worse than NL

   We believe that BP will perform better than AM because while AM and BP are both inherently unintuitive, BP requires participants to only switch between two columns, while AM requires participants to switch between columns and rows, a significantly more complex task. We believe that NL will outperform BP because it is more intuitive.

2. On the Incoming Links (IL) task, BP will perform better than AM and NL

   We believe BP will outperform AM and NL because of how BP is structured. Unlike on AM, which requires specifically focusing on a highlighted node column, BP requires the participant to only focus on the highlighted node in the right column. We believe this is more intuitive and thus should result in BP outperforming AM. Additionally, we recognized that NL would suffer from significant visual clutter, especially at high densities. It may not be possible to differentiate between incoming and outgoing nodes on a dense NL visualization, as arrows may overlap with other links. Given that BP does not contain arrows, as directional information is not displayed with these arrows but instead with columns (as explained in Section 3.4), BP does not suffer as significant from visual clutter for this specific task. As such, we believe that BP will outperform both visualizations.

3. On the Common Neighbor (CN) task, BP will perform as well as AM but worse than NL

   We believe BP will perform as well as AM. BP is more intuitive for this task than AM is; however, BP also suffers from visual clutter while AM does not. We believe that the added intuitiveness of BP in this task specifically will offset any performance loss caused by visual clutter. Specifically, we believe BP is suited to this task because it displays information so that it should be easier to find common neighbors, as we have defined them in Section 6.1.2. There is less possibility for confusion of "pass-through" nodes being considered common neighbors. However, we also believe that the inherent intuitiveness of NL will cause the NL visualization to outperform BP.

4. On the Same Group (SG) task, BP will perform slightly better than AM, and NL will perform significantly better than both.

   As discussed in Section 6.1.4, on an NL visualization, grouping nodes require finding nodes located near each other with many interconnected edges. This is not as intuitive on an AM visualization, requiring users to find clusters of links and infer the nodes in groups from there. Most AM visualizations are hierarchically clustered, causing groups to be contiguous; however, locating the boundaries of clusters can be incredibly difficult. BP faces similar challenges to AM in terms of a lack of intuitiveness. However, BP is only expanded in a single dimension, as was discussed in Section 3.4. Because BP is in one dimension instead of two, we believe this makes finding groups slightly easier on BP than AM. As such, we believe that BP will slightly outperform AM.

6.3 Study Setup

This section will cover the internal elements of the study. We will cover our decision regarding how our study mode and the networks we used in the study. We also discuss how visualizations and tasks were assigned.

6.3.1 Study Method

We considered performing a lab controlled study. We were trending toward a crowdsourced study, as it would give us the ability to send the survey to more people. On March 13, 2020, our university suspended all in-person activities, including lectures and lab work, due to the novel coronavirus pandemic. As such, we were faced with the decision to delay by an unknown amount of time to perform a lab controlled study or to move forward with a crowdsourced online survey. Given that the university provided no timeline or statement about when we can expect to be allowed to perform lab controlled studies again, we thought it prudent to abandon any attempt to run such a study [Uni20]. Given that we were already in favor of running a crowdsourced study, after the ruling that further in-person studies would not be allowed, we decided to run a crowdsourced study.

We believe that the results from a crowdsourced study will contribute to a discussion on network visualization efficacy. Specifically, we used Amazon's Mechanical Turk (mturk). We selected a sample of 20 participants per visualization, for a total of 60 participants. This number was decided by taking our budget for this experiment and calculating out fair payment for the average amount of time needed to complete the survey, giving us a total participant number of 60, which we then split into thirds. Our main limitation with the number of participants was entirely due to funding. Crowdsourcing study participants meant that given more funds, we could have easily found more participants. Our requirements on Mechanical Turk were that participants had completed at least 500 HITs (Human Intelligence Tasks, mturk parlance for any task on the platform) previously with a HIT acceptance rate of 95%. We selected these criteria to ensure that our participants were more likely to take the survey seriously. Additionally, we checked the results and removed any significant outliers that we believe were not taken in good faith, as is common practice for crowdsourced studies [APH17; Cro12].

6.3.2 Network Selection

This subsection focuses on what network generation model we selected, what network sizes and densities we selected, and our reasoning behind those decisions.

Network Generation

We attempted to find real-world data that would fit our desire to have smaller networks with varying densities. We were aware that it was highly unlikely that we would be able to find a network that fit our exact specifications; however, if we could find a network that was close, or could be modified to be close to our specifications, we would prefer to glean any benefit we could from using real-world data. We searched dataset collections from the Stanford Large Network Dataset Collection [LK14] and from KONECT, the Koblenz Network Collection [Kun13], but the majority of datasets were far too large. Our options for small datasets were either to choose an arbitrary subset of a larger dataset or select the only small datasets available which concerned the mating patterns of animals in various national parks in the United States. Both of these options would have resulted in data that had little connection to the participant, which would reduce overall data attractiveness [APH17]. Given that the only reason for choosing a real-world dataset was to increase data attractiveness, we ultimately concluded that using a generated dataset would offer us the control to design a dataset according to exact specifications without causing any significant effect on the resulting data.

The generated dataset we selected was the FARZ model covered in Section 5.4.

Network Size

We wanted a network size comparable to those used previously. Given that multiple studies have used sizes of 20, 50, and 100 nodes, we determined that this would be an adequate size to test on. Given that multiple studies have shown that increasing network size does not play a significant role in task accuracy or response time, we determined that we needed two network sizes to account for the effect of changing network size. We hope that by using network sizes common to previous studies, we will have more easily comparable results to such studies. Thus, we experimented with network sizes of 20, 50, and 100; however, upon visualizing 100 node networks, we found that the difference in perceived visual clutter between the 50 node network and the 100 node network was not as significant the comparison with the 20 node network. As such, using 50 nodes would not result in significantly different results than using 100 nodes. Given that we were specifically not using any interactivity, as discussed in section 4.3.2, for the comfort of our participants, we decided that using the 50 node network as our largest network was the best course of action. As such, our two network sizes are 20 nodes and 50 nodes.

Network Density

Similar to Ghoniem, we use three different density values. Given that density has been shown to have a much more significant effect on task accuracy and response time. As such, we want to provide three points of comparison for density. Our chosen density values are 1, 2, and 4. We briefly tested densities of 8 and 12; however, we found that these densities created far too much visual

clutter on NL and BP visualizations. Given that we do not have interactivity, as discussed in section 4.3.2, we decided that the best course of action was using the density values we had already decided on. These density values are roughly comparable to the density values used in previous studies.

6.3.3 Visualization Assignment

In our pilot survey, we had twelve participants take the survey. For the pilot survey, we had visualizations randomly assigned when the survey loaded in the browser. A participant in the pilot survey could have reloaded the browser to be assigned a different visualization. However, this was never explained to participants, and to our knowledge, no participant did this. The distribution of visualizations across the pilot study participants was seven for AM, three for NL, and three people for BP. Given these results, we ultimately determined that we needed to ensure an even distribution of visualizations across participants.

Thus, we released the survey over three days. The survey was launched at approximately the same time every day. Each day the survey was manually assigned to a specific visualization. The order we conducted was BP, NL, and then AM. We ensured that 20 participants per visualization completed the survey, and we did our best to ensure that participants took the survey in good faith. Additionally, we prevented participants from taking multiple iterations of the survey. As such, we were able to ensure that 60 unique participants took our survey.

6.3.4 Task Ordering

To remove any bias that could be caused by task order, we randomized the task order. Each overall task was performed on six different networks. We collected these six tasks into a task group. The order of the networks within a task group was randomized, and the order of the task groups was randomized as well. For example, one participant might be assigned the following task group order: Shortest Path, Common Neighbor, Incoming Link, Same Group, while another participant would be assigned a different task order. Each task group would perform the same task on the six different networks, with the order randomized, before moving onto the next task group. These random assignments were made when the survey loaded in the browser. In theory, a participant could reload the browser to get a different task order. However, this was not explained to the participant, nor was there any benefit for the participant to do this.

6.3.5 Node selection

Unlike task order, the individual node selections within the tasks needed to be consistent across the different visualizations. Given that our sample size was 20 people per visualization, the effect of having a relatively small portion of the sample of one visualization receiving "easier" tasks could invalidate the collected data. For example, for the shortest path task, if a participant in the NL visualization received two nodes that were directly adjacent, they could find a shortest path of 1. In contrast, a different participant in the AM visualization might have a different selection of highlighted nodes with a shortest path of 6, a considerably more difficult task overall. If these were performed on the same network but different visualizations, it would be impossible for us to

know if the differences in network visualization efficacy caused the difference in response time and accuracy, what we are trying to study, or if those differences were caused by random chance from highlighted node selection.

This effect could be minimized with a significantly larger sample, as it would be highly improbable that a sizable enough subset of a given visualization population sample received "easier" highlighted node selections. However, we need to ensure that this effect does not confound our data given our sample limitations. As such, the highlighted nodes are predefined, and all participants perform the same 24 tasks across the six different networks (4 tasks per network). Meaning, if a participant on AM is instructed to "find a path between node F and node A", participants on NL and BP for the same network will be given the same instructions.

To remove the effect of having a human select the highlighted nodes, the nodes for each task were decided randomly. There was no methodology to bias selecting the highlighted nodes for each task; a script was written to select nodes at random from each network.

6.4 Survey Procedure

This subsection covers what the participants did during the survey. We cover every page of the study as participants experienced them. The study was built in jsPsych [Lee14], and an example of the study procedure can be found at the following link: https://nschiele.github.io/MastersThesis/FinalSurvey.html.

Authorization

The survey was only confirmed to work on Google Chrome and Mozilla Firefox browsers. We found that it specifically did not work on Safari browsers. To ensure that participants were able to complete the survey, the first page prevented anyone from taking the survey on a non Chrome or Firefox browser. The survey might have worked on another browser, but to ensure the consistency of participants, we only allowed these two browsers to be used to take the survey.

Additionally, we wanted to ensure that participants took the survey on a computer. We had a minimum size of 700x500. If a participant attempted to take the survey with a screen size smaller than this, the authorization page would prevent them from moving forward. We can be assured that participants at least took the survey on a tablet and most likely took the survey on a computer.

Consent

Before taking the survey, we provided participants with a consent form. This form covered the rights of the participants and ensured that we have the right to store and manipulate data. We affirmed that we do not collect any personally identifiable information other than demographic information, which is optional to provide. This form ended with a checkbox to confirm acceptance and then a continue button, which only functioned if the participant ticked the confirmation checkbox.

Demographics

After the consent form, we asked for demographic information. Specifically, we asked for an age range, sex, education level, and two questions about previous experience with network analysis. These last two questions can be seen in Figure 6.1. All of these questions were multiple choice and optional, with a "prefer not to answer" option.

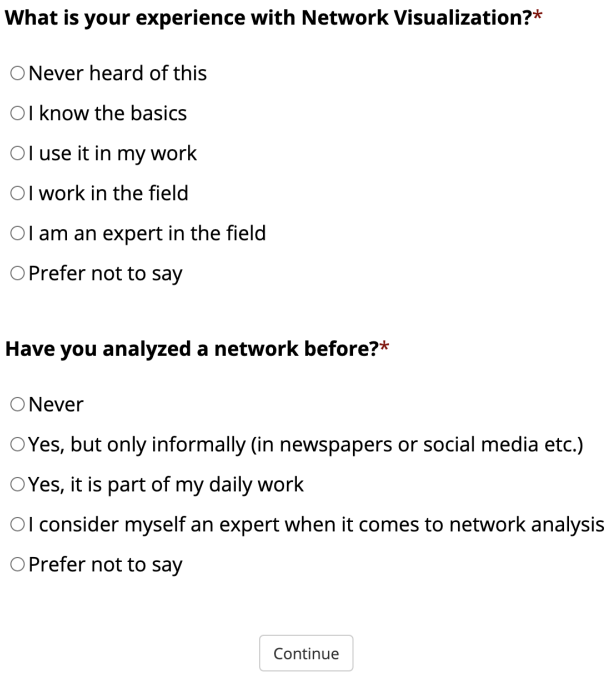


Figure 6.1: Final two questions of our request for demographic information

Visualization Introduction

Each visualization had several pages of visualization instructions. These instructions introduced the idea of network visualization on an NL visualization, as it is the most intuitive. Then for BP and AM, there were several instructional pages on how to read those visualizations. Finally, there was a comparison between BP/AM and NL, saying that these networks are the same, and the participant should convince themselves of this fact. An example of this can be seen in Figure 6.2. NL visualization instructions did not include such a page. Finally, the instructions ended with a warning that they cannot be returned to after going onto the tasks.

Task Instructions

Each task group was preceded by an instruction page specific to each task. This page explained the task that was about to be performed. It provided hints that might be relevant to a particular task or visualization, for example, reminding participants that the column of an AM visualization

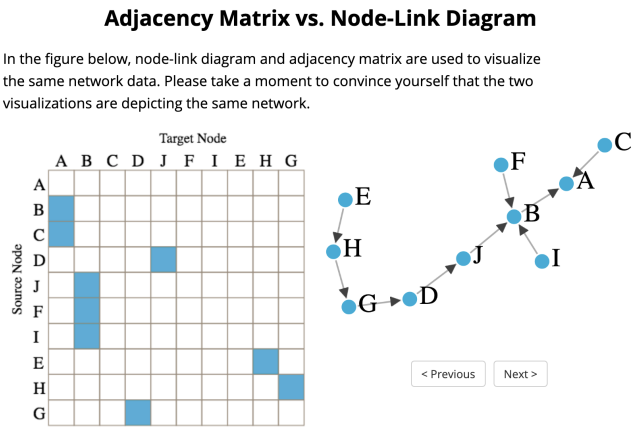


Figure 6.2: Final page of the instructions consisting of a comparison to the NL visualization

represents that target of a link. This instruction page did not include tips on how to complete the task. Additionally, this instruction page had a sample task with the answer obscured under a black bar, which would be removed on mouseover.

When the participant was ready to continue, the participant would press any button on the keyboard to begin the tasks. Figure 6.3 is an example of these task-specific instructions.

Task Evaluation

Tasks would be provided to the participant that were of the same style in the task instruction slide. Participants would then attempt to complete the task and provide an answer. For the IL, CN, and SP tasks, participants responded with a number. This was input into a number input box. A participant could enter the number with a keyboard or select up/down arrows to enter values with a mouse. This number box prevented entering values of less than zero. For SP, if there was no path, participants were instructed to enter zero. We provided similar instructions for CN and IL. For SG, we asked participants a simple yes or no question, whether two highlighted nodes were in the same group. There were two options of buttons, "Yes" or "No" that participants needed to select with a mouse.

Difficulty Evaluation

After each task on each network, we asked participants how difficult the previous task was on a 5 point Likert scale. The participant would then have the option to enter a comment if they wished to tell us something about the previous question, but this was not a requirement. After entering this information, the participant would select "Next" and be shown the subsequent task. After the tasks on all six networks were complete, the next task group's task instruction would be shown.

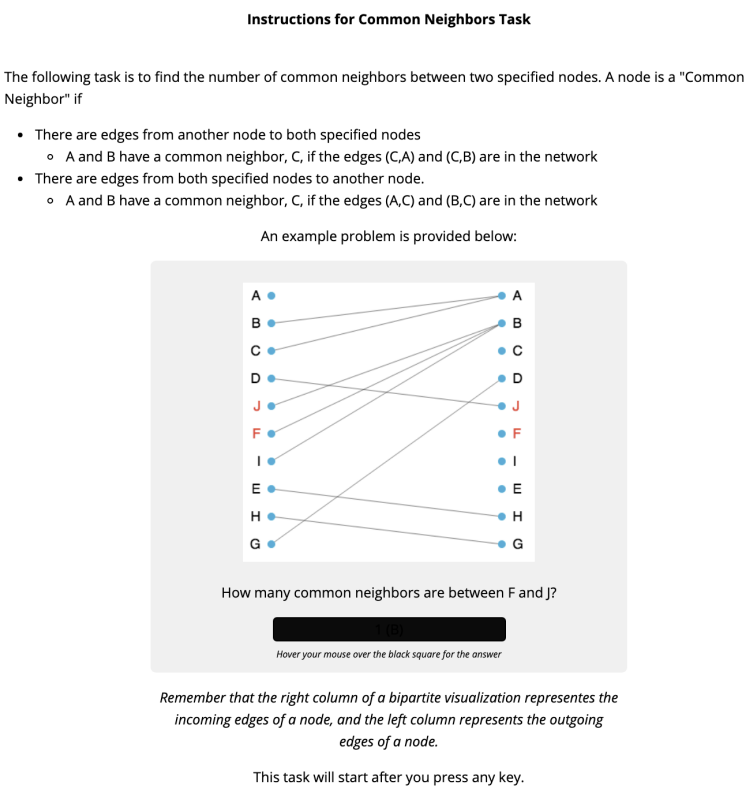


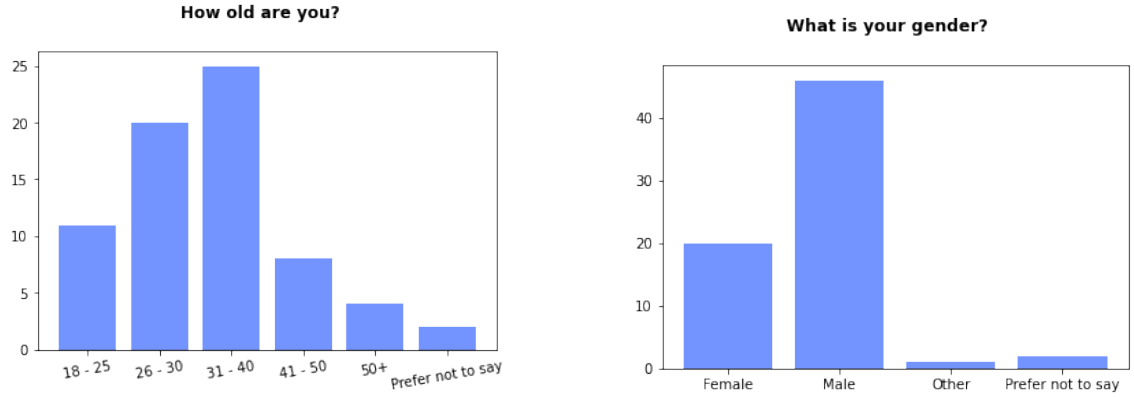
Figure 6.3: Example of task specific instructions with an example problem

7 Results

In this chapter, we cover the results of the study described in the previous section. We first cover the demographics of our study sample. We then examine the task response accuracy of each task. Finally, we examine the task response time for each task.

7.1 Demographics

In this section, we have histograms of the various demographics of our participants. Our overall goal was to assess network visualization as pertains to an average person. This is part of the reason we did not perform a lab-controlled study, as was discussed in Section 6.3.1. The demographics of our sample allow us to assess if our sample is, in fact, representative of a broader population of people.

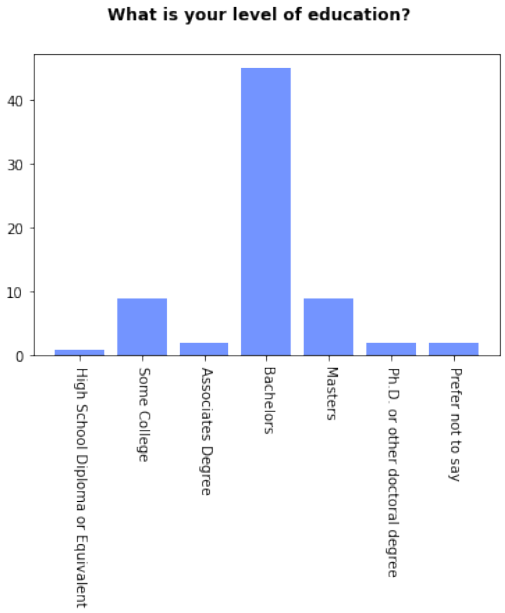


(a) Respondents organized by reported age

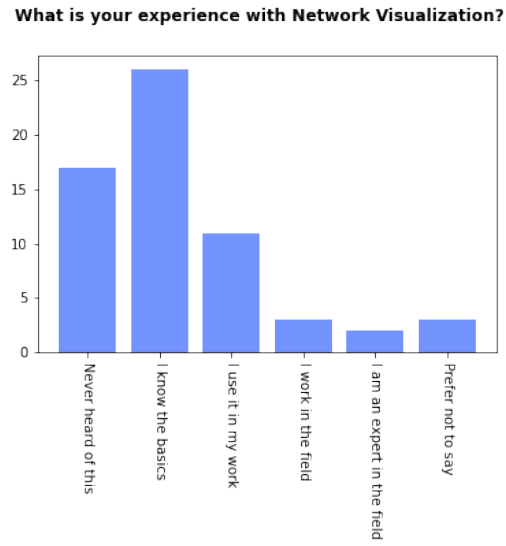
(b) Respondents organized by reported gender

Figure 7.1: Respondents organized by reported age and gender

We see in Figure 7.1a that the vast majority of respondents were between 18 and 40. We expected this result given the demographics of the population of mturk users, 88% of whom self report an age between 18 and 49. Additionally, older people tend to be less familiar with newer technology. Hence mturk being an entirely online platform, presents a barrier to entry for older people. What as unexpected with our demographics was the sex separation. Approximately 49% of mturk users are female, and yet less than a third of our respondents were female, as seen in Figure 7.1b [Hit16]. We are entirely unsure why this is the case. While there may be differences in network evaluation performance between genders, this is not what we are seeking to study. However, it is worth noting in the event this research is referred to in the future by someone looking to study gendered differences in network efficacy.



(a) Respondents organized by reported level of education



(b) Respondents organized by reported experience with network visualization

Figure 7.2: Respondents organized by reported level of education and network visualization experience

The vast majority of our respondents reported an education level of a bachelor's degree or higher, as seen in Figure 7.2a. This was roughly expected, with 51% of mturk workers self-reporting at least a bachelor's degree [Hit16]. Given that our Human Intelligence Task (HIT) was an academic survey, it is expected that it will be skewed toward those prone to take academic surveys. As such, a higher than average percentage of respondents reporting at least a bachelor's degree is expected for our sample. Additionally, we can see from Figure 7.2b that most respondents have had some experience with networks. We have no data regarding what to expect from the general population regarding network exposure. However, given that most maps are a network, it would not be an exaggeration to say most people have some experience with networks; however, they may not have recognized the connection between a road map, for instance, and a network as an abstract concept. As mentioned in Section 6.4, the demographic questions were asked before the abstract concept of a network was explained in the instructions section of the study, so it is entirely possible that these responses were skewed toward not knowing what a network is, because of the lack of a connection between the real world idea and the abstract concept, not because of an actual lack of knowledge.

Finally, we asked if participants had experience with network analysis, and, as the responses shown in the Figure 7.3 example, we see responses that are incredibly similar to those found in Figure 7.2b. This is also expected. It would be highly unusual to have responses showing experience with network analysis without exposure to network visualizations. Simultaneously, many of these responses may suffer from the same issue as previously described, where participants might not fully understand what the question is asking. It might have been prudent to move these final two demographics questions to after the instructions, to ensure that all participants had an adequate understanding of the abstract concepts being discussed before asking about their exposure to such concepts.

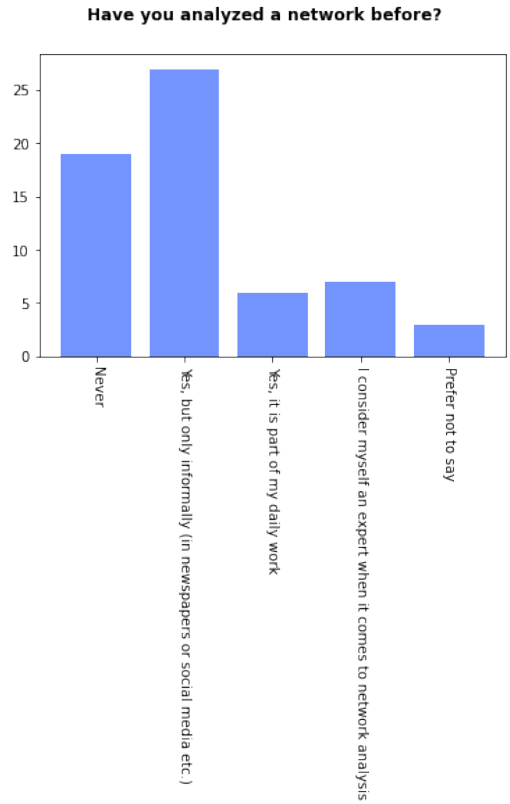


Figure 7.3: Respondents organized by reported experience with network analysis

7.2 Task Accuracy

In this section, we examine the accuracy of the participants with each task. We have organized these by size and by density as a mechanism to examine the effects of changing both the size and density on overall accuracy. As was stated in Section 6.3.5, each task used the same selection of nodes on the same network. The only difference between the task on a network of a given size and density was the visualization itself (AM/BP/NL). This could cause some individual tasks to be easier than others; however, we should see the differences between the different visualizations because we did not randomize the node selections for the tasks between visualizations. Variance caused by node selection, for example, if a specific task on a given size and density was abnormally straightforward to complete, should be balanced out by averaging tasks of a given size or density. This methodology is not novel; the same methodology was used by Ghoniem [GFC04]. For the subsequent plots of task accuracy, we show a 95% confidence interval (CI).

In Figure 7.4, we see for the Shortest Path task, AM outperformed BP and NL on the smallest network size, but on the larger network size, that all three were within the 95% CI for each visualization, so we will be unable to draw any conclusions for the SP task at the 50 node size. Moving along, we see for Incoming Links that NL outperforms BP or AM for the smallest network size, while BP outperforms AM or NL on the larger network sizes. Another noteworthy point is that the IL task organized by size shows the smallest amount of variance in the confidence intervals. On

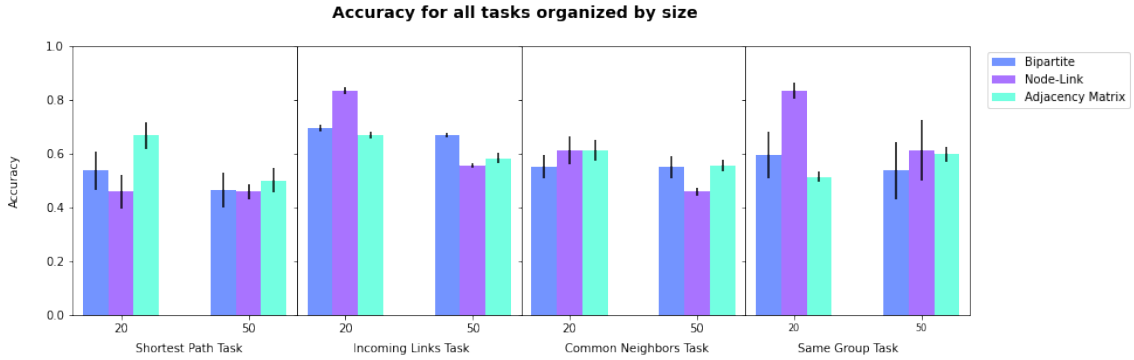


Figure 7.4: The accuracy of study participant responses averaged by network size with 95% confidence intervals

the Common Neighbor task, we see that there does not appear to be a significant difference in the accuracy across the different visualizations for small network sizes. For large networks, BP and AM seem to perform similarly, while NL performs worse. Finally, on small networks, NL visualizations do significantly better than either AM or BP, which perform similarly for the Same Group task. For large networks, there does not appear to be significant differences in the performance across varying visualizations. It is noteworthy that for large networks, the confidence interval is quite large.

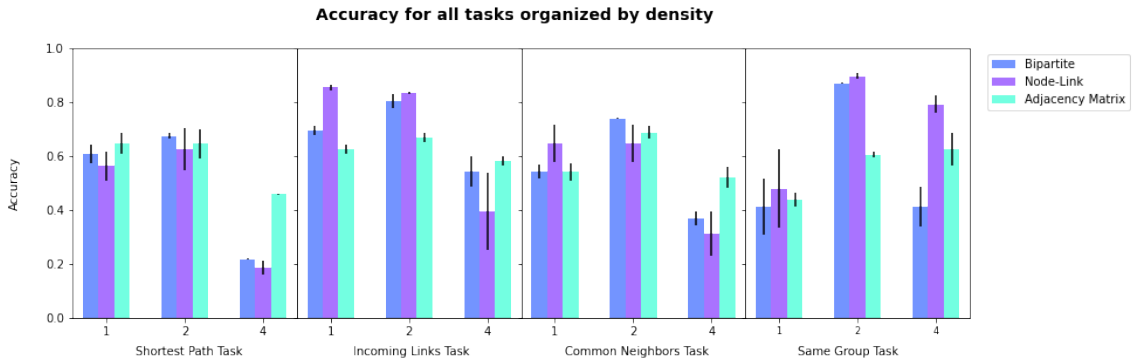


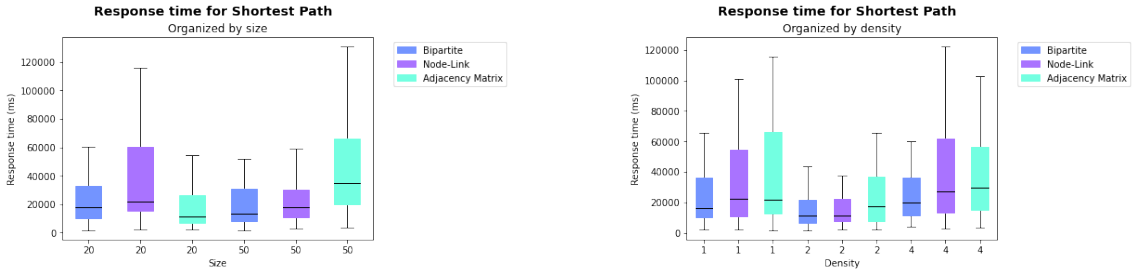
Figure 7.5: The accuracy of study participant responses averaged by network density with 95% confidence intervals

In Figure 7.5, we see for the Shortest Path task, that for a density of 1 or 2, there is no significant difference between the various visualizations. For our densest network, we find that the adjacency matrix visualization outperforms either BP or NL. Moving along, we see for Incoming Links that NL outperforms BP or AM for the smallest density, while BP and NL perform similarly better than AM for a density of 2. For the densest network, all visualizations perform similarly. It is noteworthy that IL's confidence interval on NL is significantly larger than any other confidence interval. On the Common Neighbor task, we see that all visualizations perform similarly for the least dense network. For a density of 2, BP slightly outperforms NL and AM, which perform similarly. For the

densest network, AM outperforms BP or NL. Finally, for the Same Group task, on the least dense network, all visualizations performed similarly. For networks with a density of 2, NL outperforms AM, which in turn outperforms BP.

7.3 Response Time

In this section, we examine the time taken by respondents to complete the tasks. Because we did not implement a hard time limit, as has been done on previous studies such as the one conducted by Ghoniem [GFC04], participants could take an excessive amount of time. Participants could leave their browser window open and return to the same question after a significant time. It was also possible for participants to spend a relatively excessive amount of time on any given question; however, they were not paid more for taking more time, so there was little incentive to complete the survey slowly. We did receive several outliers in our responses. We believe these outliers to be a combination of some people taking a long time to complete the survey, and some people leaving the survey open while they did other things. Given that some of the extreme outliers cause the remaining data to be muddled, we have excluded the outliers of the response time for the plots below.



(a) Oragnized by size test

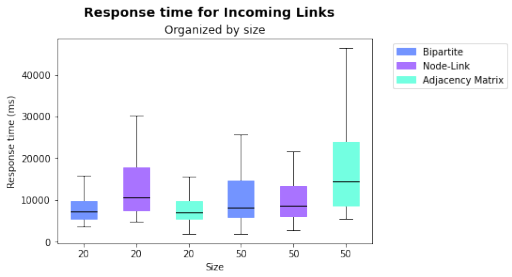
(b) Organized by Density

Figure 7.6: Boxplots of the response times on the Shortest Path task

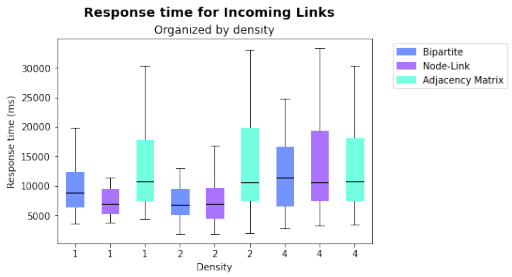
In Figure 7.6, we see that on the Shortest Path task, when organized by size, we see that NL visualizations take considerably longer than BP or AM visualizations. For larger graphs, BP and NL take similar amounts of time, while AM takes significantly longer. When organized by density, we see that we have a similar response time across all visualizations for a density of 1 and 4, with BP being slightly faster. For a density of 2, we have a shorter response time for all visualizations; however, AM is slower than BP or NL.

In Figure 7.7, we see that on the Incoming Links task organized by size, the NL visualization took participants more time on average than either BP or AM visualizations, which performed relatively equivalently. For the larger networks, AM took longer than either NL or BP, which took similar amounts of time. When organize by density, we see that AM and BP visualizations take approximately the same amount of time regardless of density. We see that NL visualizations increase in response time as density increases.

In Figure 7.8, we see that on the Common Neighbors task organized by size, we see that AM takes less time than either BP or NL. For the larger networks, BP took less time than either NL or AM, which took similar amounts of time. When organized by density, we see that for a density of 1, the

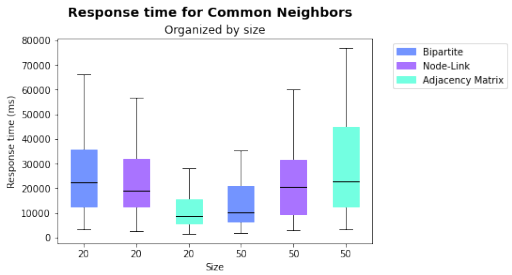


(a) Organized by size

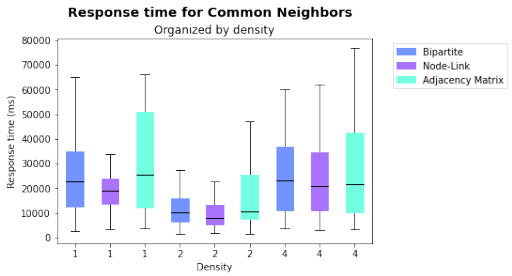


(b) Organized by Density

Figure 7.7: Boxplots of the response times on the Incoming Link task



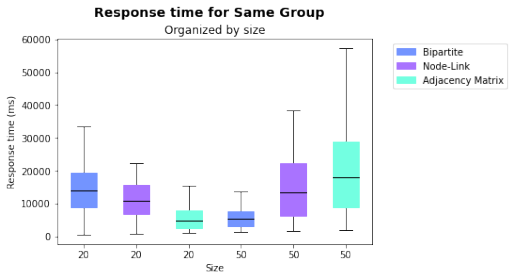
(a) Organized by size



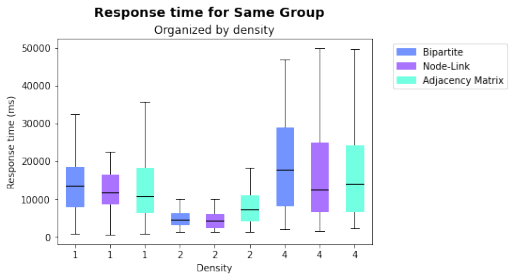
(b) Organized by Density

Figure 7.8: Boxplots of the response times on the Common Neighbor task

NL visualization, on average, takes the shortest amount of time compared to BP and AM, which take relatively similar amounts of time. For a density of 2, we see that AM generally takes longer while BP and NL take similar amounts of time. For a density of 4, we find a relatively equivalent distribution of response times across all visualizations.



(a) Organized by size



(b) Organized by Density

Figure 7.9: Boxplots of the response times on the Same Group task

In Figure 7.9, we see that on the Same Group task organized by size, we see for smaller networks, We find BP mostly takes longer than NL, which in turn takes longer than AM. This is reversed for larger networks, where BP is significantly faster than NL, which is slightly faster than AM. When organized by density, we see that for a density of 1, all visualizations take a relatively similar amount of time. For a density of 2, we see that AM generally takes longer while BP and NL take similar amounts of time. For a density of 4, we find a relatively equivalent distribution of response times across all visualizations.

8 Discussion

In this chapter, we discuss the findings of the previous chapter. We accept or reject our hypotheses and offer our interpretations of the data used to reach those conclusions. We also discuss the limitations of our study and the findings. This discussion starts with our findings and then ends with the limitations of the study as a whole.

8.1 Findings

As was stated in Section 6.1, we chose tasks that we felt would best test the differences in efficacy between the various network visualizations. We based on hypotheses in Section 6.2 on these tasks. Ultimately, we find the following:

8.1.1 Shortest Path (SP) Hypothesis

On the Shortest Path (SP) task, BP will perform better than AM but worse than NL.

We find insufficient evidence to support this hypothesis. For small networks, we see that AM outperforms NL, while BP performs in between the two visualizations. For large networks, all visualizations perform approximately equally. The most interesting aspect of this outcome is that NL is outperformed by AM, which is a result that has not been seen in previous NVE studies, especially for connectivity type tasks. The likely cause of this outcome is small, high-density graphs being significantly easier on AM while being more difficult on NL, and these two outcomes are averaged when sorting by size.

This reasoning is confirmed mainly by the accuracy plot organized by density. We see that for densities of 1 and 2, we find that all visualizations perform roughly equally, with only networks of density 4 showing an advantage to AM visualizations. This behavior could be explained by visual clutter caused by increased numbers of edges affecting BP and NL visualizations while not affecting AM visualizations. Hence the performance on AM is relatively consistent as density increases while the performance on NL and BP suffers.

In terms of response time, when organized by density, we see this odd behavior of response times decreasing between networks of density 1 and 2, and then increasing again for networks of density 4. As will be discussed in Section 8.2, this is likely due to how target nodes were selected for the task questions, the process of which is explained in Section 6.3.5. In short, because each task uses identical nodes, due to our limited sample size and wanting to be able to compare different visualizations directly, we did not randomize target nodes in the task questions. It is entirely possible that some tasks were assigned target nodes that are considerably easier to solve than others. We

would rather the difficulty of the question asked to be identical across all visualizations rather than randomize the questions to potentially get biased results. This is likely the cause of the strange behavior in the response time plot.

Overall, given that we hypothesized that BP would generally outperform the other visualizations, and in every case, BP performs on par with NL, and on par or worse than AM, we do not find sufficient evidence to support this hypothesis and thus reject it.

8.1.2 Incoming Links (IL) Hypothesis

On the Incoming Links (IL) task, BP will perform better than AM and NL.

We again do not find sufficient evidence to support this hypothesis. For smaller networks, we see that NL outperforms BP and AM, which perform similarly. This could be due to the lack of visual clutter on smaller networks and the increased intuitiveness of the NL visualization. For larger networks, we see BP outperforming NL and AM, which perform similarly. This could be caused by increased visual clutter on NL visualizations, making it more challenging to find the incoming links to an individual node. On BP, the task is more intuitive than NL, and BP is more insulated against visual clutter than NL, as nodes cannot overlap; however, as was stated in Section 3.4, BP visualizations are not immune to visual clutter, as links can still overlap.

In this vein, we do see that the performance of BP visualizations does decrease with increasing density. The same occurs on NL, with a significantly greater decrease in accuracy on networks of density 4. We see that AM does not suffer similar decreases in performance with increases in density. This behavior is consistent with our understanding of the effect of visual clutter on NL and BP graphs, as was discussed in Chapter 3. As density increases, information is lost due to visual clutter, and without interactivity to reduce the effects of visual clutter, the task becomes increasingly difficult. On AM, there is no issue with visual clutter as there are no overlapping elements. As such, the performance of AM does not appear to be affected by increasing density.

When examining the response times of the participants organized by size, we see that for BP and AM, increasing network size causes an increase in response time, though it is noteworthy that this increase is 40% greater for AM than for BP. We see a decrease in the response time on NL, and we believe that this is likely caused by people recognizing that the task is too challenging to complete and merely guessing an answer. As size increases, visual clutter on NL also increases, potentially causing information about the in-degree of a given node to be lost. If this information cannot be determined, it stands to reason that a participant would take a guess and move on. Whereas if the task is possible to complete, the participant will take the time necessary to accomplish that task.

The response times organized by density show something similar. We can see that AM takes approximately the same amount of time regardless of density, which does follow with the nature of the task on AM. Increasing the network size does cause the participant to complete more comparisons, but increasing network density does not affect this. The task is completely identical on AM networks of the same size. As such, we would not expect a variation of this take when organized by size. We see similar behavior compared to the response times organized by size on BP and NL, whereas density increases from 1 to 2, we see that response time largely decreases. We believe this could be due to visual clutter making participants guess quickly instead of taking the time to respond. However, we see that for networks with a density of 4, response times for

both BP and AM increase. This is not consistent with our reasoning behind decreasing response times. Given the methodology behind how the task nodes are determined, as stated in Section 6.3.5, and that there are only two networks for each density (a network of size 20 and 50), it is entirely possible that the task on the networks of density 4 were more straightforward than the same tasks on networks of other densities, and participants did not quickly consider the task impossible to complete. This idea is further explained in Section 8.2.

Overall, we primarily see evidence of the idea that BP is as adaptable to changing network sizes as AM, while it is equally susceptible to changes in network density as NL. We do not see the task-specific advantage we expected to see on BP, given that either NL or AM often outperforms it. As such, we reject this hypothesis.

8.1.3 Common Neighbor (CN) Hypothesis

On the Common Neighbor (CN) task, BP will perform as well as AM but worse than NL.

We find little evidence to support this hypothesis. For small networks, all visualizations fall within confidence intervals of each other. As such, there is no significant evidence to suggest that one visualization outperforms the others. For larger networks, NL performs worse than AM or BP, which performs within the confidence intervals of each other. This is contrary to what we expected to see and is evidence toward rejecting the hypothesis.

When looking at the networks organized by density, we see NL outperform AM or BP for density 1. For density 2, BP outperforms NL or AM. Finally, for density 4, we see AM outperform NL or BP. The behavior could be explained by visual clutter. As was discussed in Chapter 3, both NL and BP suffer from visual clutter on high-density networks due to the overlapping of links. The similarity of the responses on AM across variations of density and size seem to indicate the finding common neighbors on AM is a task that does not increase in difficulty with size or density; while completing this task on BP or NL may be impossible due to visual clutter.

If we compare the response times across the different visualizations, we can see that AM universally takes longer than the other visualizations. We believe that because finding neighbors on AM is always possible, participants take the time to complete the task; while on BP or NL, when there is significant visual clutter, participants are more likely to quickly guess an answer rather than spend time attempting to find the correct answer as finding a correct answer may prove to be an impossible task.

Overall, our results showed that, on average, BP performed about as well as AM, with AM having a slight improvement in accuracy. Our expectation that NL would prove to be the most effective visualization did not materialize. Overall, we find limited evidence to support this hypothesis, and thus largely reject it.

8.1.4 Same Group (SG) Hypothesis

On the Same Group (SG) task, BP will perform slightly better than AM, and NL will perform significantly better than both.

We find that there is some evidence to suggest that this hypothesis should be accepted. For small networks, the results are exactly what we expected to see, with NL having significantly better accuracy than either BP or AM, and BP having slightly better accuracy than AM but within the 95% confidence bounds. For large networks, we only see a slight improvement of NL over BP and AM, but all visualization accuracies are within the confidence intervals, so we do not see the significant performance improvement of NL over BP or AM.

When looking at networks by density, we see for the least dense networks, there is no significant difference in the accuracy across the different visualizations. For a density of 2, we see that NL has significantly outperformed both AM and BP. We also see that BP has very significantly outperformed AM, which we did not expect. Finally, for the densest networks, we see that NL outperforms both BP and AM, with AM significantly outperforming BP.

It is worth noting that the SG task was unique in that it was the only yes-no task, with all the other tasks requiring a positive integer response. If all participants randomly guess their answer, the expected value of the response accuracy should be approximately 50%, with half of responses being correct. As such, responses near or below 50% can largely be attributed to random guessing, as can be seen in the response accuracy of all three visualizations on the least dense networks. This tells us that respondents were mostly unable to distinguish groups on any visualization on low-density networks, consistent with the findings in previous NVE studies [OJK18; RMO+19]. Given that groups are defined mainly by frequent connections to smaller subsets of nodes within a network, the lack of significant numbers of connections in low-density networks would contribute to an inability to distinguish different groups.

When looking at the task response time, we see that for small networks, on BP visualizations, participants took significantly longer to respond than participants on AM visualization. This relationship was exactly reverse on large networks. Across varying densities, we see that for networks of density 2, we see a decrease in response time. As was discussed in the previous section, this could either be due to the methodology behind node selection, or it could be caused by participants recognizing that with the lost information due to visual clutter, the task is either incredibly difficult or impossible to complete.

Overall, in terms of accuracy, we mainly see what we expected, on networks where participants could distinguish groups (dense networks), NL visualizations outperform BP or AM. However, the distinction between the efficacy of BP and AM is not clear. As such, we can tentatively accept that NL outperforms BP or AM for the Same Group task.

8.1.5 Comparison with other NVE studies

We have several notable counter examples to the consensus of the findings of previous NVE studies. For example, we see that on the shortest path task, for small networks, we see that AM clearly outperforms NL, which is in opposition to the general consensus of NL performing better than AM on connectivity tasks, especially on smaller networks. This could be the result of a confounding effect, such as those found by Okoe when they discovered that many survey participants were quickly answering potentially due to fatigue or due to the difficulty of the question; this behavior caused their results to have a few counter examples[OJK18]. As was discussed in Section 6.3.1, we

did our best to limit confounding variables, but given our sample size limitations and the nature of crowdsourced studies, it is entirely possible that we do not fix confirming results entirely due to a confounding effect [APH17].

Another potential explanation for our counter examples is the lack of interactivity. It could be the case that a lack of interactivity causes a smaller reduction in efficacy on AM than it does for NL. We generally see larger decreases in expected performance on NL than we do on AM, so there is evidence in our results to suggest that this is the case. Of course, given that we did not specifically test for interactivity, given that we did not have a visualization with interactive elements, we cannot draw any conclusion on this theory.

8.1.6 Interactivity

We did not include a hypothesis about the lack of interactivity as we were not explicitly testing interactivity; all of our visualizations did not include any interactive elements. However, we can still compare our response accuracy to the response accuracy of similar tasks asked by previous NVE studies.

We see a stark decrease in accuracy on our visualization as compared to other NVE studies. Ghoniem had responses for most tasks near or above 90%, while our average is 55% [GFC04]. The one potential cause for this is that Ghoniem sampled experts in their study. As was discussed in Chapter 4, all of the participants in the Ghoniem study had postgraduate degrees in mathematics or computer science. These experts already had familiarity with graph theory, thus were not being newly introduced to the concept. Our study, in contrast, had 20% of participants with no previous experience with graph theory or network theory. However, Okoe included interactive elements and had a crowdsourced sample, and only saw an approximately 20% drop in performance [OJK18]. Given that our sample was roughly comparable to the Okoe sample, it is unlikely that sample selection alone can result in the 40% drop in task accuracy that we see compared to Ghoniem.

The likely cause of our significant decrease in task accuracy is the lack of any interactive elements. Interactive elements such as highlighting nodes and links on mouseover can reduce the information lost due to visual clutter. Highlighting on mouseover can draw attention to visual elements overlaid by other visual elements, allowing participants to see information that would otherwise not be available because that information is lost due to visual clutter. As such, especially for high-density networks on AM and BP visualizations, we see significantly worse performance on the same task than other NVE studies. Some tasks are not possible because critical information, such as entire edges, are overlaid by other visual elements. Visualizations with interactive elements that mitigate this issue should not see a significant performance drop.

We can mostly confirm that interactive elements affect network visualization efficacy, as our task accuracy was considerably lower than that of similar tasks on other NVE studies. However, we did not specifically measure the effect of interactivity on visualization efficacy, as this would be an entirely different experiment, comparing two otherwise identical visualizations and tasks, one with various interactive elements, and a control without such interactive elements.

8.2 Limitations

As was discussed in Section 6.3.5, we selected the nodes for each task at random before the study was launched. Given that participants would only complete tasks on a single visualization, we needed to ensure that we would be able to compare the performance across different visualizations. This decision was informed by our limited sample size, which was caused by budgetary constraints, as explained in Section 4.5.2. We specifically did not select the task targets manually to avoid bias being introduced by human task selection.

The result, however, is that some tasks may be more difficult than others. It is possible that, for example, the SP task on the network of size 20 with density 2 (network 2) is significantly more difficult than the same task on the network with 50 nodes and density 4 (network 6), simply because of target node selection. As such, we can draw limited conclusions as a result of changing size and density. However, because all tasks were performed on the exact same targets for the exact same networks, we can directly compare the results from different visualizations on the same network.

One method we could have instituted was to randomly select targets from a small list of possible targets that was pre-generated. With this methodology, we would have had, for example, four different possible targets for each task and network, which would be assigned at random to participants. Even without small sample size, it would have likely resulted in a greater ability to compare results from the same visualization on different networks. However, doing this on such a small sample would have likely caused an increase in the size of our confidence intervals, which for some tasks and some visualizations, are already relatively large. This would have also reduced our ability to draw conclusions of results between the different visualizations, which was our primary goal.

This is not to say that our testing methodology was flawed. We deliberately chose to select task targets in this manner in order to enable the comparison we wished to make. Given a larger budget, and thus larger sample size, we likely would have made different decisions to enable a broader range of comparisons.

9 Conclusion and Future Work

This chapter summarizes our findings as they apply to network efficacy evaluation tasks and network usage more generally. We also cover our findings concerning interactivity. Additionally, we suggest about how to expand this research in the future.

9.1 Conclusion

Our results mostly agree with those found by previous NVE studies in terms of AM and NL visualizations. In general, for connectivity or cluster related tasks, NL is still the preferred visualization. For network property type tasks, as defined in Section 4.4.2, there is an advantage for AM visualizations. We find that for connectivity tasks, BP performs on par with NL visualizations. We find that for connectivity tasks, there is little difference between the accuracy of BP versus the accuracy of NL. Further, we find that the performance of BP on cluster-related tasks is on par with AM visualizations. However, we did find several counter examples which do not agree with these results. As was discussed in Section 8.1.6, we believe that many of these counter examples are the result of our not using any interactive elements in our visualizations. Further work will be needed to confirm this hypothesis, as described in Section 9.2.

One central theme that has been repeated throughout this work is the idea of visual clutter or information lost due to the organization of visualization elements. As was discussed in Chapter 3, NL can suffer from overlapping nodes and edges, BP can suffer from overlapping edges, and AM does not suffer from visual clutter. This does not make AM a superior visualization, as there are several negatives to AM visualizations that do not necessarily apply to NL or BP, such as understandability.

As was consistent with the findings of Ghoniem, NL visualizations generally have a decreasing performance with increasing network size and density [GFC04]. We have largely confirmed these results, finding that increasing network size and density will generally cause a decrease in performance on NL visualization task accuracy. It stands to reason that BP accuracy would suffer with increasing density, but not with increasing network size, as nodes cannot overlap in a BP visualization. This is largely what we have found, that for large, sparse networks, BP visualizations generally outperform NL visualizations. For dense networks of any kind, the overlapping links on NL and BP visualizations cause significant loss of information and result in worsening task accuracy. As such, for dense networks without interactivity, AM visualizations are the optimal visualization to use.

Our study did not include any interactivity, which largely contributed to the significantly worse performance across every task compared to previous NVE studies, even on those studies that did not sample experts [OJK18; RMO+19]. Interactivity does have the effect of increasing accuracy, likely through increasing understandability. Not including interactivity results in decreasing network efficacy, a conclusion that has primarily been considered obvious.

Critically, the lack of interactivity had a more substantial effect on NL visualizations than on AM visualizations. That is to say, the reduction in accuracy and increase in response time is more apparent on NL than on AM when comparing to previous NVE studies. As was discussed in Section 8.1.6, this is likely caused by interactivity being able to compensate for visual clutter. As such, especially for dense networks, interactive elements are necessary to allow the network to be effective, understandable, and usable.

9.2 Future Work

Future work will be needed to further establish the relationship between BP, AM, and NL, similar to how this relationship has already primarily been defined between AM and NL. This work would need to use a larger variety of task types to further refine our understanding. As was explained in Section 8.2, our limited use of different task types may result in evidence toward a relationship that is not the case.

As for interactivity, Further work is needed to define which interactive elements increase efficacy. As was stated in Chapter 3, previous studies have used "common" interactive elements, but there has not been study into which interactive elements have which effects and whether or not there are interactive elements that potentially have an adverse effect on task accuracy and response time. The structure of this work could significantly resemble the study we performed, only with fewer or no variations in basic network properties and instead variations in interactive elements, ranging from no interactivity as we examined here to a full suite of interactive tools similar to those implemented by Nobre [NWHL20].

Bibliography

[AABB11] Y. Y. Ahn, S. E. Ahnert, J. P. Bagrow, A. L. Barabási. "Flavor network and the principles of food pairing". In: *Scientific Reports* (2011) (cit. on p. 25).

[AB02] R. Albert, A. L. Barabási. "Statistical mechanics of complex networks". In: *Reviews of Modern Physics* (2002), pp. 47–97 (cit. on p. 46).

[APH17] D. Archambault, H. Purchase, T. Hoßfeld. *Evaluation in the Crowd. Crowdsourcing and Human-Centered Experiments*. 2017, pp. 154–190 (cit. on pp. 30–32, 35, 37, 38, 40, 41, 43, 52, 53, 69).

[BBDW14] F. Beck, M. Burch, S. Diehl, D. Weiskopf. "The State of the Art in Visualizing Dynamic Graphs". In: *Proceedings State of the Art Reports (STARs)* (2014), pp. 83–103 (cit. on p. 36).

[BETT94] G. D. Battista, P. Eades, R. Tamassia, I. G. Tollis. "Algorithms for drawing graphs: an annotated bibliography". In: *Computational Geometry: Theory and Applications* (1994), pp. 235–282 (cit. on p. 23).

[BEW95] R. A. Becker, S. G. Eick, A. R. Wilks. "Visualizing network data". In: *IEEE Transactions on Visualization and Computer Graphics* (1995), pp. 16–28 (cit. on p. 36).

[BGT13] D. Bannister Michael J.and Eppstein, M. T. Goodrich, L. Trott. "Force-Directed Graph Drawing Using Social Gravity and Scaling". In: *Graph Drawing*. 2013, pp. 414–425 (cit. on p. 24).

[BLLW76] N. Biggs, E. Lloyd, L. Lloyd, R. Wilson. *Graph Theory 1736-1936*. Clarendon Press, 1976 (cit. on pp. 17, 18).

[BMK96] J. Blythe, C. McGrath, D. Krackhardt. "The effect of graph layout on inference from social network data". In: *Lecture Notes in Computer Science* (1996), pp. 40–51 (cit. on pp. 30–33, 35, 37).

[BPA+07] C. Baranauskas, P. Palanque, J. Abascal, S. Barbosa, R. Bernhaupt, M. Winckler, D. Navarre. *Human-Computer Interaction*. 2007, pp. 412–424 (cit. on p. 38).

[Con20] F. Connections. *Flight Connections Route map S7 Airlines*. 2020. URL: https://www.flightconnections.com/route-map-s7-airlines-s7 (visited on 08/10/2020) (cit. on p. 22).

[Cro12] K. Crowston. "Amazon Mechanical Turk: A Research Tool for Organizations and Information Systems Scholars". In: *Shaping the Future of ICT Research. Methods and Approaches*. Ed. by A. Bhattacherjee, B. Fitzgerald. Springer Berlin Heidelberg, 2012, pp. 210–221 (cit. on pp. 41, 52).

[DS13] C. Dunne, B. Shneiderman. "Motif Simplification: Improving Network Visualization Readability with Fan, Connector, and Clique Glyphs". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2013, pp. 3247–3256 (cit. on p. 37).

[DSD16] A. Debiasi, B. Simões, R. De Amicis. *Schematization of Clutter Reduction Techniques in Geographic Node-link Diagrams using Task-based Criteria*. 2016, pp. 107–114 (cit. on p. 24).

[ER59] P. Erdös, A. Rényi. "On random graphs I." In: *Publicationes Mathematicae* (1959), pp. 290–297 (cit. on pp. 43, 44).

[Eul41] L. Euler. "Solutio Problematis ad Geometriam Situs Pertinentis, 1736". In: *Commentarii academiae scientiarum Petropolitanae* (1741), pp. 128–140 (cit. on pp. 17, 18).

[FARZ18] J. Fagnan, A. Abnar, R. Rabbany, O. R. Zaiane. *Modular Networks for Validating Community Detection Algorithms*. 2018 (cit. on pp. 46, 47).

[GFC04] M. Ghoniem, J. D. Fekete, P. Castagliola. "A comparison of the readability of graphs using node-link and matrix-based representations". In: *IEEE Symposium on Information Visualization* (2004), pp. 17–24 (cit. on pp. 29–37, 39–41, 49, 51, 61, 63, 69, 71).

[GS93] D. Gries, F. B. Schneider. *A Logical Approach to Discrete Math (Monographs in Computer Science)*. 1993 (cit. on p. 33).

[Hit16] P. Hitlin. "Research in the Crowdsourcing Age, a Case Study". In: *Pew Research Center* (2016), pp. 1–7 (cit. on pp. 59, 60).

[Hu09] Y. Hu. *Visualizing graphs with node and edge labels*. 2009 (cit. on pp. 26, 28).

[JRHT14] R. Jianu, A. Rusu, Y. Hu, D. Taggart. "How to display group information on node-link diagrams: An evaluation". In: *IEEE Transactions on Visualization and Computer Graphics* (2014), pp. 1530–1541 (cit. on pp. 28, 35, 36, 43).

[KEC06] R. Keller, C. M. Eckert, P. J. Clarkson. "Matrices or node-link diagrams: Which visual representation is better for visualising connectivity models?" In: *Information Visualization* (2006), pp. 62–76 (cit. on pp. 29–35, 37, 39–41, 49, 51).

[KN02] M. Kawabe, Y. Nimura. "LA-13 Fast Random Generation Method for Connected Graphs (A. Algorithm/Basic)". In: *Information Technology Letters*. Aug. 2002, pp. 25–26 (cit. on pp. 31, 34).

[KN04] M. Kawabe, Y. Nimura. *ISPT Random Graph Server*. https://web.archive.org/web/20050414042137/http://www.ispt.waseda.ac.jp/~kawabe/rgs/index.shtml. Waseda University, 2004 (cit. on pp. 31, 35).

[Kob12] S. Kobourov. *Spring Embedders and Force Directed Graph Drawing Algorithms*. 2012 (cit. on p. 24).

[Kun13] J. Kunegis. "KONECT: The Koblenz Network Collection". In: *Proceedings of the 22nd International Conference on World Wide Web*. Association for Computing Machinery, 2013, pp. 1343–1350 (cit. on p. 53).

[LBI+12] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, S. C. Empirical. "Empirical Studies in Information Visualization: Seven Scenarios". In: *IEEE Transactions on Visualization and Computer Graphics* (2012), pp. 1520–1536 (cit. on p. 32).

[LBM78] E. K. Lloyd, J. A. Bondy, U. S. R. Murty. "Graph Theory with Applications". In: *The Mathematical Gazette* (1978), p. 63 (cit. on pp. 18, 23).

[Lee14] J. R. de Leeuw. “jsPsych: A JavaScript library for creating behavioral experiments in a Web browser”. In: *Behavior Research Methods* (Mar. 2014), pp. 1–12 (cit. on p. 55).

[Lex20] A. Lex. *How Far Can We Push Crowdsourced Evaluation of Visualization Techniques?* July 2020 (cit. on p. 38).

[LK14] J. Leskovec, A. Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. http://snap.stanford.edu/data. June 2014 (cit. on p. 53).

[Mel06] G. Melancon. “Just how dense are dense graphs in the real world?: A methodological note”. In: *Proceedings of BELIV'06: BEyond time and errors - novel EvaLuation methods for Information Visualization. A workshop of the AVI 2006 International Working Conference* (2006) (cit. on pp. 33, 34).

[NC88] T. Nishizeki, N. Chiba. “Planar Graphs: Theory and Algorithms”. In: *Annals of Discrete Mathematics* (1988), pp. 1–232 (cit. on pp. 23, 24).

[NWHL20] C. Nobre, D. Wootton, L. Harrison, A. Lex. *Evaluating Multivariate Network Visualization Techniques Using a Validated Design and Crowdsourcing Approach*. 2020 (cit. on pp. 22, 30, 31, 33–39, 41, 49, 72).

[OJK18] M. Okoe, R. Jianu, S. Kobourov. “Revisited experimental comparison of node-link and matrix representations”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2018), pp. 287–302 (cit. on pp. 30–37, 39–41, 49, 51, 68, 69, 72).

[RMO+19] D. Ren, L. R. Marusich, J. O'Donovan, J. Z. Bakdash, J. A. Schaffer, D. N. Cassenti, S. E. Kase, H. E. Roy, W. Y. Lin, T. Höllerer. “Understanding node-link and matrix visualizations of networks: A large-scale online experiment”. In: *Network Science* (2019), pp. 242–264 (cit. on pp. 30–35, 37, 39–41, 49, 51, 68, 72).

[Tur84] G. Turán. “On the succinct representation of graphs”. In: *Discrete Applied Mathematics* (1984), pp. 289–294 (cit. on p. 24).

[Uni20] University of Stuttgart. *Newsticker: Information on the coronavirus*. Mar. 2020. URL: https://www.uni-stuttgart.de/en/university/news/corona/ (cit. on p. 52).

[WF94] S. Wasserman, K. Faust. *Social network analysis: Methods and applications*. Cambridge University Press, 1994 (cit. on p. 25).

[Wik08] Wikimedia Commons. *International E Road Network*. 2008. URL: https://commons.wikimedia.org/wiki/File:International_E_Road_Network_green.png (cit. on p. 14).

[Wik17] Wikimedia Commons. *Barabasi Albert Graph*. 2017. URL: https://commons.wikimedia.org/wiki/File:Barabasi_albert_graph.svg (cit. on p. 45).

[Wil96] R. J. Wilson. *Introduction to Graph Theory*. Fourth Edi. Addison Wesley Longman Limited, 1996 (cit. on pp. 18, 24, 33).

[WS98] D. J. Watts, S. H. Strogatz. “Collective dynamics of ‘small-world’ networks”. In: *Nature* (June 1998), pp. 440–442 (cit. on pp. 44, 45).

[XRP+12] K. Xu, C. Rooney, P. Passmore, D. H. Ham, P. H. Nguyen. “A user study on curved edges in graph visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* (2012), pp. 2449–2456 (cit. on p. 23).

[YN06] B. Yost, C. North. “The perceptual scalability of visualization”. In: *IEEE Transactions on Visualization and Computer Graphics* (2006), pp. 837–844 (cit. on p. 32).

All links were last followed on October 15, 2020.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

---

place, date, signature