

Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Masterarbeit

Evaluating Adjacency Matrix for Network Visualization

Raman Sharma

Course of Study: Informatik
Examiner: Prof. Dr. Daniel Weiskopf
Supervisor: Moataz Abdelaal, M.Sc.

Commenced: October 23, 2020
Completed: April 23, 2021

Abstract

Adjacency Matrix (AM) is one of the commonly used techniques to visualize networks. While an AM provides a clean and compact representation for dense networks, several studies have shown that it is not suitable for path-related tasks. Several visualization techniques have been proposed to address this limitation. This thesis goes in the same direction by investigating the influence of the rotation of the matrix and the visual encoding of links on the user performance at different network analysis tasks. To this end, a crowd-sourced user study was conducted to evaluate different variants of AM across several network analysis tasks and different network properties. The results reveal that the accuracy is not significantly affected by the rotation of the matrix, as well as the visual encoding of links, across all tasks. This means that users achieve similar accuracy on all visualizations, indifferent of the rotation of the matrix or the visual encoding of links. Several isolated cases exist, where both, the VCD and ArcM record a visibly higher accuracy than the AM. However, the difference in accuracy is deemed not significant in all those cases. For example, visibly higher accuracy is recorded on the VCD as well as on the ArcM than on the AM for the task concerning the detection of mirror symmetry, across most networks. However, the difference in accuracy is deemed not significant across those networks. The results also reveal that the answer time is a critical factor. The rotation of the matrix, as well as the visual encoding of links, lead to significant degradation of answer time across most connectivity-related tasks. As for the path-finding task, the answer time is not significantly affected by both, the rotation of the matrix and the visual encoding of links.

Contents

1	Introduction	15
1.1	Motivation	15
1.2	Problem Definition	17
1.3	Contributions	17
1.4	Outline	18
2	Related Work	19
2.1	Comparison of Adjacency Matrix with Node-Link Diagram	19
2.2	Proposal and Evaluation of Variants of Adjacency Matrix	20
3	Variants of Adjacency Matrix	23
3.1	Adjacency Matrix (AM)	23
3.2	Vertical Cross-Diagram (VCD)	25
3.3	Horizontal Cross-Diagram (HCD)	27
3.4	Single Curved Polyline Adjacency Matrix (SCPAM)	28
3.5	Arc Diagram (ArcD)	29
3.6	Arc Matrix (ArcM)	30
3.7	Variants Considered for the Study	31
4	Study Design	33
4.1	Tasks and Hypotheses	33
4.2	Independent Variables	37
4.3	Dependent Variables	38
4.4	Confounding Factors	38
4.5	Experimental Design	39
4.6	Study Conduction	40
5	Network Data	43
6	Results	51
6.1	Count Neighbors (CN)	51
6.2	Identify Link Endpoints (ILE)	52
6.3	Bidirectional Linkage (BL)	53
6.4	Direct Connection (DC)	55
6.5	Find Path (FP)	56
7	Discussion	63
7.1	Discussion of the Results	63
7.2	Challenges	66

8 Conclusion	69
A Calculation of P-values for Network Property Influence Testing	71
A.1 Count Neighbors (CN)	71
A.2 Identify Link Endpoints (ILE)	73
A.3 Bidirectional Linkage (BL)	76
A.4 Direct Connection (DC)	78
A.5 Find Path (FP)	80
B Calculation of P-values for Hypothesis Significance Testing	83
B.1 Count Neighbors (CN)	84
B.2 Identify Link Endpoints (ILE)	86
B.3 Bidirectional Linkage (BL)	88
B.4 Direct Connection (DC)	90
B.5 Find Path (FP)	92
C Demographics	95
D Data concerning Vision Deficiency	99
Bibliography	101

List of Figures

1.1	A directed network visualized as a Node-Link Diagram.	15
1.2	A directed network visualized as an Adjacency Matrix.	17
3.1	A directed network visualized as an Adjacency Matrix.	23
3.2	Illustration of an Adjacency Matrix with node “4” and its corresponding row and column highlighted.	24
3.3	A directed network visualized as a Vertical Cross-Diagram.	26
3.4	Illustration of a Vertical Cross-Diagram with node “4” and its corresponding “cross” highlighted.	26
3.5	A directed network visualized as a Horizontal Cross-Diagram.	27
3.6	An illustration of the “cross” corresponding to node “4” in a Horizontal Cross-Diagram.	28
3.7	A directed network visualized as a Single Curved Polyline Adjacency Matrix.	28
3.8	A directed network visualized as an Arc Diagram.	29
3.9	A directed network visualized as an Arc Matrix.	30
3.10	Extending a link between two nodes in an Arc Matrix.	31
3.11	A directed network visualized using different techniques.	32
4.1	A flow chart showing how participants proceed from task to task.	40
5.1	A flow chart depicting the steps for generating directed networks.	43
5.2	A flow chart depicting the steps for displaying the clusters in a visualization for a network.	46
5.3	A dendrogram depicting the clusters in a network.	47
5.4	Illustration of the effect of a reordering process.	48
5.5	Two possible ways of depicting a link in a directed network are shown. The upper right half of the figure shows the two possible unidirectional links between two nodes, while the lower right half of the figure shows a bidirectional link between two nodes.	48
6.1	Illustration of the accuracy achieved on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for CN task. The network increases in size/density from left to right/top to bottom.	53
6.2	Illustration of the distribution of answer time recorded on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for CN task. The network increases in size/density from left to right/top to bottom.	54
6.3	Illustration of the accuracy achieved on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for ILE task. The network increases in size/density from left to right/top to bottom.	55

6.4	Illustration of the distribution of answer time recorded on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for ILE task. The network increases in size/density from left to right/top to bottom.	56
6.5	Illustration of the accuracy achieved on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for BL task. The network increases in size/density from left to right/top to bottom.	57
6.6	Illustration of the distribution of answer time recorded on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for BL task. The network increases in size/density from left to right/top to bottom.	58
6.7	Illustration of the accuracy achieved on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for DC task. The network increases in size/density from left to right/top to bottom.	59
6.8	Illustration of the distribution of answer time recorded on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for DC task. The network increases in size/density from left to right/top to bottom.	60
6.9	Illustration of the accuracy achieved on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for FP task. The network increases in size/density from left to right/top to bottom.	61
6.10	Illustration of the distribution of answer time recorded on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for FP task. The network increases in size/density from left to right/top to bottom.	62
C.1	Distribution of the participants based on their age.	95
C.2	Distribution of the participants based on their gender.	95
C.3	Distribution of the participants based on their education.	96
C.4	Distribution of the participants based on their level of experience with network visualization.	96
C.5	Distribution of the participants based on their level of knowledge regarding network analysis.	97
C.6	Distribution of the participants based on their familiarity with the term "Adjacency Matrix".	97
D.1	Distribution of the participants based on their performance in the color-blindness test.	99
D.2	Distribution of the participants based on vision deficiencies they potentially suffer from.	99

List of Tables

4.1	List of network analysis tasks considered for the study.	36
4.2	List of hypotheses for how the visualizations AM, VCD, and ArcM will compare with each other for each task, split by accuracy and answer time	37
4.3	The six networks of varying complexity used for the study.	38
A.1	Accuracy achieved on each visualization across each network for CN task.	71
A.2	Difference in accuracy achieved on one visualization across two networks, respectively, varying in size for CN task. Significant difference (according to Welch's t-test) highlighted in bold.	71
A.3	Difference in accuracy achieved on one visualization across two networks, respectively, varying in density for CN task. Significant difference (according to Welch's t-test) highlighted in bold.	72
A.4	Mean answer time recorded on each visualization across each network for CN task.	72
A.5	Difference in answer time recorded on one visualization across two networks, respectively, varying in size for CN task. Significant difference (according to Welch's t-test) highlighted in bold.	72
A.6	Difference in answer time recorded on one visualization across two networks, respectively, varying in density for CN task. Significant difference (according to Welch's t-test) highlighted in bold.	73
A.7	Accuracy achieved on each visualization across each network for ILE task.	73
A.8	Difference in accuracy achieved on one visualization across two networks, respectively, varying in size for ILE task. Significant difference (according to Welch's t-test) highlighted in bold.	74
A.9	Difference in accuracy achieved on one visualization across two networks, respectively, varying in density for ILE task. Significant difference (according to Welch's t-test) highlighted in bold.	74
A.10	Mean answer time recorded on each visualization across each network for ILE task.	74
A.11	Difference in answer time recorded on one visualization across two networks, respectively, varying in size for ILE task. Significant difference (according to Welch's t-test) highlighted in bold.	75
A.12	Difference in answer time recorded on one visualization across two networks, respectively, varying in density for ILE task. Significant difference (according to Welch's t-test) highlighted in bold.	75
A.13	Accuracy achieved on each visualization across each network for BL task.	76
A.14	Difference in accuracy achieved on one visualization across two networks, respectively, varying in size for BL task. Significant difference (according to Welch's t-test) highlighted in bold.	76

A.15	Difference in accuracy achieved on one visualization across two networks, respectively, varying in density for BL task. Significant difference (according to Welch's t-test) highlighted in bold.	76
A.16	Mean answer time recorded on each visualization across each network for BL task.	77
A.17	Difference in answer time recorded on one visualization across two networks, respectively, varying in size for BL task. Significant difference (according to Welch's t-test) highlighted in bold.	77
A.18	Difference in answer time recorded on one visualization across two networks, respectively, varying in density for BL task. Significant difference (according to Welch's t-test) highlighted in bold.	77
A.19	Accuracy achieved on each visualization across each network for DC task.	78
A.20	Difference in accuracy achieved on one visualization across two networks, respectively, varying in size for DC task. Significant difference (according to Welch's t-test) highlighted in bold.	78
A.21	Difference in accuracy achieved on one visualization across two networks, respectively, varying in density for DC task. Significant difference (according to Welch's t-test) highlighted in bold.	78
A.22	Mean answer time recorded on each visualization across each network for DC task.	79
A.23	Difference in answer time recorded on one visualization across two networks, respectively, varying in size for DC task. Significant difference (according to Welch's t-test) highlighted in bold.	79
A.24	Difference in answer time recorded on one visualization across two networks, respectively, varying in density for DC task. Significant difference (according to Welch's t-test) highlighted in bold.	79
A.25	Accuracy achieved on each visualization across each network for FP task.	80
A.26	Difference in accuracy achieved on one visualization across two networks, respectively, varying in size for FP task. Significant difference (according to Welch's t-test) highlighted in bold.	80
A.27	Difference in accuracy achieved on one visualization across two networks, respectively, varying in density for FP task. Significant difference (according to Welch's t-test) highlighted in bold.	80
A.28	Mean answer time recorded on each visualization across each network for FP task.	81
A.29	Difference in answer time recorded on one visualization across two networks, respectively, varying in size for FP task. Significant difference (according to Welch's t-test) highlighted in bold.	81
A.30	Difference in answer time recorded on one visualization across two networks, respectively, varying in density for FP task. Significant difference (according to Welch's t-test) highlighted in bold.	81
B.1	Difference in accuracy achieved on two different visualizations, respectively, across each network for CN task. Significant difference (according to Welch's t-test) highlighted in bold.	84
B.2	Difference in answer time recorded on two different visualizations, respectively, across each network for CN task. Significant difference (according to Welch's t-test) highlighted in bold.	85

B.3	Difference in accuracy achieved on two different visualizations, respectively, across each network for ILE task. Significant difference (according to Welch's t-test) highlighted in bold.	86
B.4	Difference in answer time recorded on two different visualizations, respectively, across each network for ILE task. Significant difference (according to Welch's t-test) highlighted in bold.	87
B.5	Difference in accuracy achieved on two different visualizations, respectively, across each network for BL task. Significant difference (according to Welch's t-test) highlighted in bold.	88
B.6	Difference in answer time recorded on two different visualizations, respectively, across each network for BL task. Significant difference (according to Welch's t-test) highlighted in bold.	89
B.7	Difference in accuracy achieved on two different visualizations, respectively, across each network for DC task. Significant difference (according to Welch's t-test) highlighted in bold.	90
B.8	Difference in answer time recorded on two different visualizations, respectively, across each network for DC task. Significant difference (according to Welch's t-test) highlighted in bold.	91
B.9	Difference in accuracy achieved on two different visualizations, respectively, across each network for FP task. Significant difference (according to Welch's t-test) highlighted in bold.	92
B.10	Difference in answer time recorded on two different visualizations, respectively, across each network for FP task. Significant difference (according to Welch's t-test) highlighted in bold.	93

List of Algorithms

5.1	Generator(n, m, k)	44
5.2	Connect(i, N, C)	45
5.3	Add_Directions_To_Links(N, s)	49
5.4	Add_Missing_Links(N, prev_num_of_links, current_num_of_links)	50

1 Introduction

1.1 Motivation

We are experiencing rapid growth in the volume of relational data. Exploring, analyzing, and understanding large and complex relational data has become an increasingly challenging task. Therefore, relational data is modeled by using networks. A network is an abstract structure that is composed of nodes connected by links. Nodes represent entities, such as individuals and organizations. Links represent the relations between nodes, such as contact and friendship. For example, the World Wide Web is a huge network of websites connected by hyperlinks [AB02]. Networks are increasingly encountered in numerous fields of research and application, such as biology (e.g., protein-protein interactions [dF10]), economics (e.g., formal and informal organizations [MB09]) and sociology (e.g., friendship networks [Kad12]). A network is either undirected or directed. In an undirected network, the links between nodes have no directions associated with them. In contrast, in a directed network, the links between nodes have a direction associated with them. For example, a Twitter network can be regarded as a directed network, because a person can follow several people, but they (i.e., the followed ones) do not have to follow that person (i.e., the follower) back.

Networks are visualized, to get a picture of the relationships between the nodes. In other words, a network structure is displayed that gives more apprehension about the existing connections in the network. A wide variety of network visualization techniques exist that can be used to visualize a network. The most common way to visualize a network is to depict the nodes as circles and the links as lines or curves. Each node is labeled with the name of its corresponding entity. Such a type of visualization is known as *Node-Link Diagram (NLD)*. Figure 1.1 shows a small, directed network visualized as a NLD.

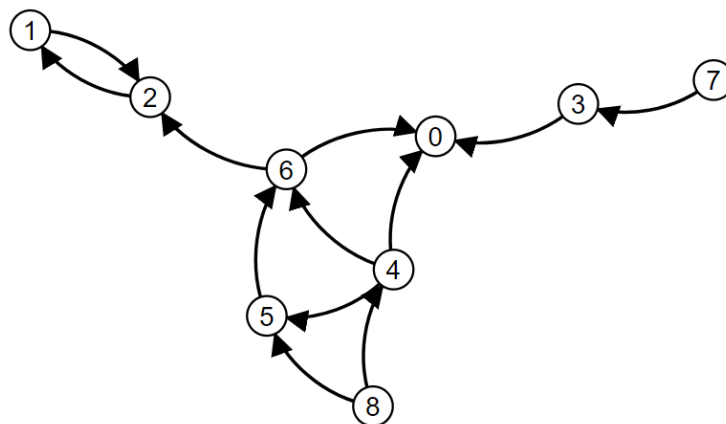


Figure 1.1: A directed network visualized as a Node-Link Diagram.

Instead of simple lines or curves, the links are depicted as curved arrows that indicate the direction of the relationships between the nodes. This type of visualization provides a deep understanding of the existing connections between the nodes in the underlying network.

A variety of network analysis tasks can be performed on the NLD to gain interesting insights into the underlying network. Lee et al. derived taxonomy of tasks that are common in the field of network analysis [LPP+06]. According to the task taxonomy described in [LPP+06], several topology-based tasks can be performed on the NLD in Figure 1.1, for example:

1. How many nodes are accessible from node “6”?
2. Which nodes are accessible from node “4” and node “8”?
3. Does a path from node “1” to node “7” exist?

In the context of directed, real-world networks (e.g., Twitter network), the three tasks listed above can be formulated as follows:

1. How many people does person X follow?
2. Which people are followed by person X and person Y?
3. Does a “Follow-Chain” starting at person X and ending at person Y exist?

If the network is small and sparse, then the tasks listed above can be performed with ease on the NLD. However, if the network increases in size and density, then the problems of link-crossings and overlapping nodes occur that lead to an increase in visual clutter [KEC06]. In turn, visual clutter leads to a degradation of performance at some tasks [RLMJ05]. For example, the first two tasks listed above will become more difficult to perform on the NLD as the underlying network grows in size and density. Therefore, a NLD is not always a preferred visualization choice for large and dense networks. A variety of algorithmic approaches exists for creating a NLD that adheres to a subset of aesthetic rules (e.g., minimization of link-crossings and even distribution of nodes) [BRSG07] so that visual clutter is reduced to a minimum in dense networks. For example, one approach positions the nodes of a network along the perimeter of a circle [BB04], while another approach models nodes and links of a network as forces [FK13]. Even though these approaches achieve the implementation of a subset of aesthetic criteria, other aesthetic criteria are violated to such an extent that the advantages and disadvantages of these approaches cancel each other out, and thus, visual clutter remains a problem. Since visual clutter is increased by link-crossings and overlapping nodes, another way to reduce it is to apply a visualization technique that maps nodes and links to visual variables other than circles and lines, respectively. Indeed, such a visualization technique exists that visualizes the network as an *Adjacency Matrix (AM)*. Figure 1.2 depicts the network from Figure 1.1, now, visualized as an AM.

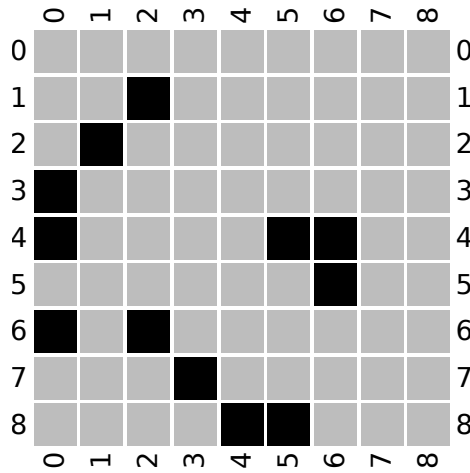


Figure 1.2: A directed network visualized as an Adjacency Matrix.

The network is represented with a $V \times V$ matrix, where V is the number of nodes in a network. The nodes are depicted on both the X-axis (i.e., horizontal axis) and the Y-axis (i.e., vertical axis). Each cell, or figuratively square, depicts a link. If a link between two nodes exists, then the corresponding cell is colored black. The rows in the matrix display all outgoing links, while the columns display all incoming links. This type of visualization scales well for dense networks while preserving a clean layout, and therefore, visual clutter is reduced to a minimum [GFC04].

1.2 Problem Definition

Even though visual clutter is reduced to a minimum, several user studies [GFC04; KEC06; OJK18; RMO+19] have shown that users struggle to perform path-related tasks on an AM. Therefore, the AM is regarded as an unsuitable visualization choice for path-related tasks. In just the same way as different visualization techniques were proposed to address visual clutter in a NLD, a variety of visualization techniques [HF07; HFM07; SBPP15] were proposed to address the limitation of the AM for path-related tasks. To some extent, the proposed approaches mitigated the AM's unsuitability for path-related tasks. However, when they [HF07; HFM07; SBPP15] compared their respective visualization techniques with the NLD for a variety of network analysis tasks, the respective results showed that either the NLD remained a more well-suited visualization choice for path-related tasks or that the difference in user performance recorded on the respective visualizations at the said tasks was not significant enough to declare any of the visualization techniques more well-suited than the others.

1.3 Contributions

In this work, reasons for the AM's unsuitability for path-related tasks are identified. Solutions to the problems are proposed with the introduction of several variants of AM. Each of the variants implements at least one solution to a problem. At the same time, the proposed variants preserve the

AM's capability to represent dense networks. Two variants of AM are proposed with the focus on the rotation of the traditional matrix. These variants are named *Vertical Cross-Diagram* (VCD) and *Horizontal Cross-Diagram* (HCD) (see Sections 3.2 and 3.3, respectively). Another variant of AM is proposed with the focus on the visual encoding of links. This variant is named *Single Curved Polyline Adjacency Matrix* (SCPAM) (see Section 3.4). The last two variants are proposed with the focus on the rotation of the matrix as well as the visual encoding of links. These variants are named *Arc Diagram* (ArcD) and *Arc Matrix* (ArcM) (see Sections 3.5 and 3.6, respectively). The goal of this thesis is to compare two of the proposed variants, namely, VCD and ArcM, with the AM for several network analysis tasks, to investigate how the rotation of the matrix and the visual encoding of links influence the user performance at those tasks, particularly path-related tasks. The comparison of the visualizations is accomplished by conducting a crowd-sourced study, where participants are exposed to one of the visualizations (i.e., AM, VCD, or ArcM) on that they perform a variety of network analysis tasks across networks differing in size and density. Contrarily to most of the studies discussed in this thesis (see Chapter 2), directed networks, which comprise properties of real-world networks, are visualized to gain interesting, possibly new scientific insights.

1.4 Outline

This thesis is structured as follows:

Chapter 2 – Related Work This chapter covers studies concerning the AM. While in some studies the evaluation of the AM was carried out in comparison with the NLD, in other studies variants of AM were proposed that were then compared with either the AM, NLD, or both.

Chapter 3 - Variants of Adjacency Matrix In this chapter, the reasons for the AM's unsuitability for path-related tasks are laid out. The issues are addressed with the introduction of different variants of AM. At the end of the chapter, it is discussed which AM variants are considered for the study.

Chapter 4 - Study Design This chapter details everything that comes with the development and conduction of the study, such as the definition of tasks, the development of the hypotheses, the measures collected during the study, the experimental design, and lastly, the study procedure.

Chapter 5 - Network Data In this chapter, the procedure for generating the networks that are relevant for the study is described.

Chapter 6 - Results This chapter illustrates how well users performed the network analysis tasks on the visualizations across different networks. The significance of the results is determined by applying statistical analysis methods.

Chapter 7 - Discussion In this chapter, the results from Chapter 5 are evaluated. The results are put into perspective for hypotheses derived in Chapter 4. Lastly, challenges are described that were faced during the conduction of the study.

Chapter 8 - Conclusion The conclusion sums up the outcome of the thesis. Also, a future outlook on this topic is provided.

2 Related Work

In the past, a variety of studies were conducted to evaluate the AM for a set of network analysis tasks across one or more networks. Section 2.1 covers studies that were carried out to evaluate the AM in comparison with the NLD. Section 2.2 covers works that proposed one or more variants of AM. In some of those works, a study was conducted to evaluate their proposed variant (or variants) in comparison with either the AM, NLD, or both.

2.1 Comparison of Adjacency Matrix with Node-Link Diagram

Ghoniem et al. [GFC04] conducted a study, where users performed seven topology-based tasks on the AM and NLD. To achieve good and meaningful results, each task was performed on several undirected networks differing in size and density. A limited number of interactions were allowed that could be applied on both visualizations, such as hovering over a node or link and selecting a node or link. When the user hovered or selected a node, not only the node itself but also its incident links were highlighted in a different color. Likewise, when the user hovered or selected a link, not only the link itself but also its endpoints were highlighted in a different color. The evaluation results showed that the NLD was more well-suited than the AM for small and sparse networks. However, on larger and denser networks, users achieved more accurate results on the AM than they did on the NLD. The path-finding task proved to be challenging to perform on the AM, whereas the NLD was a good fit for this task except for the largest network. Keller et al. [KEC06] came to a similar conclusion. The comparison of both visualizations was carried out for six topology-based tasks across directed, real-world networks differing in size and density. The interactive capabilities that could be applied on both visualizations were similar to the ones in [GFC04], albeit hovering over a node or link was not provided as an option. The evaluation results confirmed the results in [GFC04]. Okoe et al. [OJK18] conducted a large-scale, crowd-sourced study to add more to the evaluation results discovered in the previous studies. In this study, each user performs 14 tasks on one of the visualizations. Both visualizations depicted one, comparatively large, undirected, real-world network. The interactive capabilities in [GFC04] were extended by making it possible to select multiple nodes on both visualizations. Furthermore, zooming and panning were possible on both visualizations. The evaluation results largely confirmed the findings in [GFC04]. Moreover, users achieved similar accuracy (i.e., number of correct answers on average) and answer time on both visualizations for cluster-based tasks, although the AM was more well-suited than the NLD for estimating the number of clusters. On the flip side, the user performance on the NLD was better than the user performance on the AM for memorability-based tasks. So far, only those studies were discussed that merely focused on the topology of the underlying network. Nobre et al. [NWHL20] conducted a study, where users performed 16 attribute-based tasks either on the AM or on NLD across undirected, multivariate networks differing in size and density. For the realization of the multivariate data in the networks, several node- and link attribute encodings

were applied. In the NLD, the numerical node-attribute was encoded with the circle size, and the categorical node-attribute was encoded with color. For tasks that involved more than two node-attributes, nested bar charts were used for numerical values and colored glyphs were used for categorical values. A legend was included that displayed the meaning of all visual encodings that were applied in the visualization. The numerical link attribute was encoded with link thickness and the categorical link attribute was encoded with color. In the AM, node-attributes were encoded by juxtaposing a tabular representation with the AM, keeping the rows consistent between the AM and the table. Thus, spatial region and color were used to encode categorical data, while bars were used to encode numerical attributes. Numerical link attributes were encoded with color, while categorical link attributes were encoded with brightness/saturation. On both visualizations, it was possible to search for a node, highlight one node or its neighboring nodes, and enable tool-tips. On the NLD, it was also possible to move nodes, while the AM could be sorted interactively by node label, the neighborhood of a node, or any attribute. According to Nobre et al. [NWHL20], the sorting feature added in the AM was more powerful than moving nodes on the NLD. The evaluation results showed that the AM was more well-suited than the NLD for cluster-based tasks, tasks that benefited from sorted attributes, and tasks that required scanning over the entire network. However, the NLD was more well-suited than the AM for path-related tasks and tasks involving neighboring nodes. For tasks involving link attributes, Nobre et al. [NWHL20] concluded that the AM was at least as well-suited as the NLD, even for very sparse networks. Ren et al. [RMO+19] conducted a large-scale, crowd-sourced study, where each user performs twelve topology-based tasks and four attribute-based tasks either on the NLD or on one of the two sorting variants of an AM, namely *MatrixDegree* and *MatrixGroup*, across undirected, real-world, social networks differing in size and density. In *MatrixDegree*, nodes could be sorted by their respective degree, i.e., the number of links connected to the nodes. In *MatrixGroup*, nodes could be sorted by an attribute. For all the visualizations, interactions included hovering over a node or a link. When a user hovered over a node, not only the node itself and its label were highlighted in a different color, but also its neighboring nodes and their respective labels were highlighted in a different color. Additionally, in the NLD, the stroke of the corresponding links was increased, while in both sorting variants the row and column corresponding to the node was shaded. When a user hovered over a link, not only the two connected nodes and their labels were highlighted in a different color, but also their respective neighboring nodes and labels were highlighted in a different color. Additionally, in the NLD, the stroke of the hovered link was increased, while in both sorting variants the rows and columns corresponding to the two connected nodes were shaded. Except for the tasks that involved sorting the nodes, the evaluation results showed that users were more accurate and faster when performing tasks on the NLD than on any of the sorting variants. The difference in user performance – concerning accuracy and answer time – recorded on the NLD and two sorting variants, respectively, decreased for the larger network. All in all, Ren et al. [RMO+19] concluded that the NLD recorded a better user performance.

2.2 Proposal and Evaluation of Variants of Adjacency Matrix

Henry et al. [HF07] addressed the AM’s unsuitability for path-related tasks by proposing an augmented matrix visualization, called *MatLink*. *MatLink* is an AM whose inside borders are augmented with links connecting nodes. When selecting a node in the rows or columns section, it was highlighted in a different color. Additionally, the shortest path was drawn between the selected

node and any node that was currently under the mouse pointer in a different color. Thus, paths were pre-attentively visible, making path-finding very easy. Henry et al. [HF07] conducted a study, where users performed four topology-based tasks and one cluster-based task on MatLink, the AM or the NLD, across undirected, real-world, social networks differing in size and density. To run a fair comparison with MatLink, both the AM and NLD were enhanced with the interactive features existing in MatLink. It was not specified, how these features were implemented for both visualizations. The evaluation results showed that users achieved higher accuracy on MatLink than they did on any of the other two visualizations for the majority of tasks. Unsurprisingly, users achieved higher accuracy on NLD than they did on the AM for path-related tasks. However, not only did the majority of users prefer MatLink over the AM and NLD for path-related tasks, but they also achieved a significantly higher accuracy on MatLink than they did on the AM for the said tasks. Moreover, for one path-related task which involved the search for the shortest path, users achieved higher accuracy on MatLink than they did on the NLD. Overall, MatLink proved to be a reasonable visualization choice for path-related tasks. In contrast to the previous studies, Sansen et al. [SBPP15] proposed three variants of an AM with the focus on improving the visual encoding of the links in an AM. Note that the AM is referred to as *Encoding 1* in [SBPP15]. In the first variant, namely *Encoding 2*, only existing links are rendered, meaning that if a link between two nodes exists, then the corresponding cell is filled with color, else the cell is not rendered at all. Thus, visual complexity is reduced. The second variant, namely *Encoding 3*, is a hybrid visualization of an AM and a NLD. A network is still represented with a $V \times V$ matrix with the nodes being depicted on both the X-axis (horizontal) and the Y-axis (vertical). However, each link is encoded with a polyline which connects the source node depicted on the X-axis and the target node depicted on the Y-axis. At the intersection of the source node's corresponding row and the target node's corresponding column, the polyline bends orthogonally. A small square is placed on top of the intersection point, to distinguish the bend from other polylines crossing. In the last variant, namely *Encoding 4*, each link is encoded in the same way as in Encoding 3, but instead of small squares being placed on top of the intersection point, the polyline is curved. Sansen et al. [SBPP15] argue that the curve shape helps to distinguish the bend from other polylines crossing. Sansen et al. [SBPP15] conducted a study, where each user performs one task either on one of the three AM variants or on the AM across undirected networks differing in size and density. The task consisted of selecting links one by one to find a path between two highlighted nodes within a time limit. Sansen et al. [SBPP15] chose to visualize the undirected networks as half-matrices because the pilot study revealed that displaying a link in both directions introduced visual clutter, and thus, reduced the readability of the matrices. A limited number of interactions could be applied on all visualizations, such as selecting a link, zooming, and panning on a visualization. The evaluation results showed that users achieved the highest accuracy on Encoding 3. Users achieved the lowest accuracy on Encoding 1. Encoding 3 was preferred by the majority of users while Encoding 4 was the least preferred. Although in both visualizations, Encoding 3 and Encoding 4, the links are encoded with polylines, Sansen et al. [SBPP15] concluded that the distinction between the orthogonal bend of a polyline and other polylines crossing is easier to address with shapes rather than curves. There also exist other works that addressed the AM's unsuitability for path-related tasks by proposing variants of the visualization. However, none of those works attempted to evaluate their respective variants. Nonetheless, their works are worth mentioning, because the development of AM variants plays an important part in this thesis. In addition to MatLink, Henry et al. [HFM07] proposed another AM variant, called *NodeTrix*. NodeTrix is a hybrid visualization of a NLD and an AM. It uses a NLD to visualize a network but replaces dense parts that show community structures by an AM. In other words, relationships in-between a community are modeled as an AM while

relationships between communities are modeled as NLDs. Therefore, real-world, social networks can be visualized as NodeTrix, as those types of networks exhibit community structures. Henry et al. [HF06] proposed yet another AM variant, called *MatrixExplorer*. As the title of their work says, *MatrixExplorer* visualizes a network using two visualizations, namely NLD and AM, each on a separate window. The AM is shown on one side, synchronized with the NLD on the other side. Although *MatrixExplorer* provides an effective visualization of a network, it requires the user to have two large monitors to exhaust the tools that come with the visualization, such as interactive filtering, clustering functions, reordering of the matrix, and so on.

This thesis goes in the same direction as the works discussed in Section 2.2 by conducting a crowd-sourced study to compare proposed variants of AM with the AM. Users perform a variety of network analysis tasks on one of the visualizations across networks differing in size and density (see Chapter 4). Instead of proposing a novel hybrid approach such as *MatLink* [HF07] or *NodeTrix* [HFM07], the variants proposed in this thesis can be regarded as “true derivations” of an AM that mitigate the limitation of an AM for path-related tasks (see Chapter 3). Furthermore, in this thesis, the networks are not only directed, but they also follow the evolution patterns and characteristics observed in real-world networks (see Section 5).

3 Variants of Adjacency Matrix

This chapter provides an introduction to the AM and its variants. The AM and its variants were rendered using D3.js ¹, a JavaScript library for visualizing data. First, in Section 3.1, an introduction to the *Adjacency Matrix* (AM) is provided. This section specifically highlights the problems that become evident when performing path-related tasks on an AM. Next, several variants of AM are introduced to address the highlighted issues laid out in the previous section. Sections 3.2 and 3.3 cover the variants *Vertical Cross-Diagram* (VCD) and *Horizontal Cross-Diagram* (HCD), respectively, that are proposed with the focus on the rotation of the traditional matrix. Then, Section 3.4 provides an overview of the *Single Curved Polyline Adjacency Matrix* (SCPAM) that is proposed with the focus on the visual encoding of links. Sections 3.5 and 3.6 provide an introduction to the variants *Arc Diagram* (ArcD) and *Arc Matrix* (ArcM), respectively, that are proposed with the focus on the rotation of the matrix as well as the visual encoding of links. Lastly, Section 3.7 concludes which of the variants are considered for the study.

3.1 Adjacency Matrix (AM)

An *Adjacency Matrix* (AM) is a square matrix ($V \times V$) that is used to visualize a network. Figure 3.1 shows a small, directed network visualized as an AM.

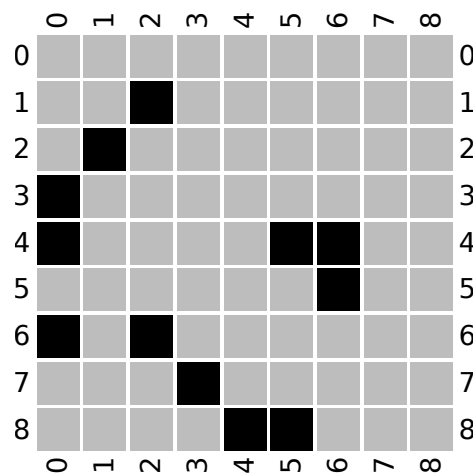


Figure 3.1: A directed network visualized as an Adjacency Matrix.

¹<https://d3js.org/>

3 Variants of Adjacency Matrix

The nodes are depicted on both the X-axis and the Y-axis. Each cell, or figuratively square, depicts a link. If a link between two nodes exists, then the corresponding cell is colored black. The rows in the matrix display all outgoing links, while the columns display all incoming links.

Users can perform several network analysis tasks on an AM. While some of the tasks can be performed with ease, others are more challenging to perform on the said visualization. For example, a task that involves counting the neighbors of a node can be easily performed on the AM [GFC04; KEC06; OJK18; RMO+19]. Figure 3.2 highlights the areas that are considered for counting the neighbors of a node in an AM (node “4” in the example).

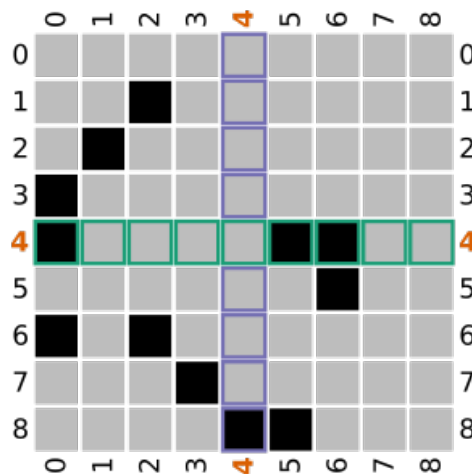


Figure 3.2: Illustration of an Adjacency Matrix with node “4” and its corresponding row and column highlighted.

To count all the neighbors of node “4”, users have to count its outgoing links, i.e., the black squares in its corresponding row, as well as its incoming links, i.e., the black squares in its corresponding column. In the end, users can conclude that node “4” has four neighbors. An AM is well-suited for this task because the procedure of counting neighbors is straightforward.

However, path-related tasks have been proven to be difficult to perform on the AM [GFC04; KEC06; OJK18; RMO+19]. For example, if the task requires users to check if a path from source node “4” to target node “2” exists in Figure 3.1, they have to perform the following steps in a best-case scenario:

1. Go to the row corresponding to node “4” depicted on the Y-axis (starting position)
2. Check if an outgoing link to node “2” (depicted on the X-axis) exists by scanning over the row corresponding to node “4”
 - 2.1. If link exists, path exists
 - 2.2. If link does not exist, select any node that has an incoming link from node “4” (i.e., “0”, “5”, or “6”)
 - 2.3. If no link exists, path does not exist
3. Go to the row corresponding to node “6” depicted on the Y-axis

4. Check if an outgoing link to node “2” (depicted on the X-axis) exists by scanning over the row corresponding to node “6”
 - 4.1. If link exists, path exists
 - 4.2. If link does not exist, select any node that has an incoming link from node “6” (i.e., “0” or “2”)
 - 4.3. If no link exists, path does not exist
5. Path found: “4” → “6” → “2”

The process listed above highlights the problems that become apparent when performing path-related tasks on an AM, which are:

- Orientating in the matrix is very laborious
- Visual encoding of links not intuitive

The first issue is caused by the depiction of the nodes on the horizontal and vertical axis, respectively. Displaying the nodes in such a way makes orientating in the matrix very laborious, as users have to continuously go back and forth between the rows and columns, which can be especially difficult for long paths or large networks. The constant change in orientation can lead to great confusion so that users become frustrated quickly with the task. In the example, users have to change their orientation from one point to another plenty of times to see if a simple connection between two nodes exists. The second issue concerns the visual encoding of links. Depicting links between nodes with squares is not intuitive, because a path between two nodes can not be easily traced by following the corresponding links in the AM. This also means that such kind of task cannot be approached using a strategy, besides the “trial and error” technique.

3.2 Vertical Cross-Diagram (VCD)

The *Vertical Cross-Diagram* (VCD) is proposed as a solution to the problem of laborious orientation in an AM. Figure 3.3 shows a small, directed network visualized as a VCD.

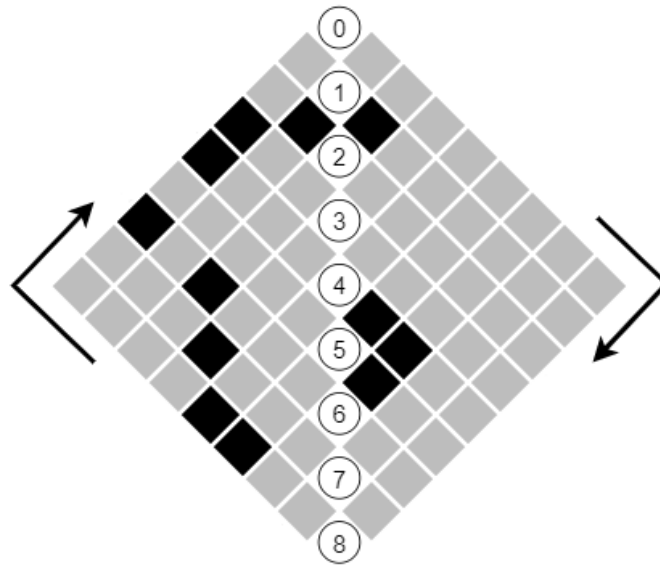


Figure 3.3: A directed network visualized as a Vertical Cross-Diagram.

The traditional matrix is rotated 45 degrees clockwise. By rotating the matrix 45 degrees, the diagonal entries are now used as a central vertical axis to represent the nodes. To emphasize this change, the diagonal entries of the matrix are encoded with circular nodes. Now, a network is represented with a diamond-shaped grid rather than a square matrix. The grid cells are used to depict the connections between the nodes. Suppose that node X is always above node Y on the central vertical axis, then, the right half of the grid depicts all outgoing links from node X to node Y, while the left half of the grid depicts all incoming links to node X from node Y. In other words, the direction of the links is clockwise, as indicated by the arrows on both sides of the visualization in Figure 3.3.

The term “Cross-Diagram” results from the visual perception of the links corresponding to each node. Each node has a corresponding “cross” depicting its outgoing and incoming links. Figure 3.4 depicts a “cross” corresponding to node “4”.

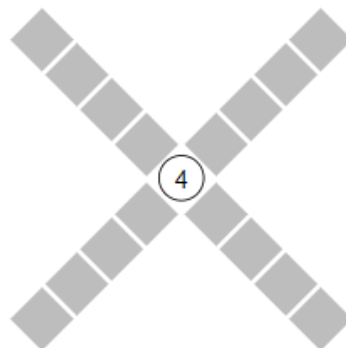


Figure 3.4: Illustration of a Vertical Cross-Diagram with node “4” and its corresponding “cross” highlighted.

The “cross” corresponding to node “4” has four arms. The upper-left and lower-right arms depict the outgoing links of node “4”, while the upper-right and lower-left arms depict the incoming links of node “4”. Obviously, not all “crosses” are “even-sized”. Also, the “crosses” corresponding to the first and last node have only two arms, such as node “0” and node “8”.

With the rotation of the matrix, orientating in the matrix happens more conveniently, because users have to move back and forth only on the central vertical axis that depicts the nodes.

3.3 Horizontal Cross-Diagram (HCD)

The *Horizontal Cross-Diagram* (HCD) is proposed as a solution to the problem of laborious orientation in an AM. Figure 3.5 shows a small, directed network visualized as a HCD.

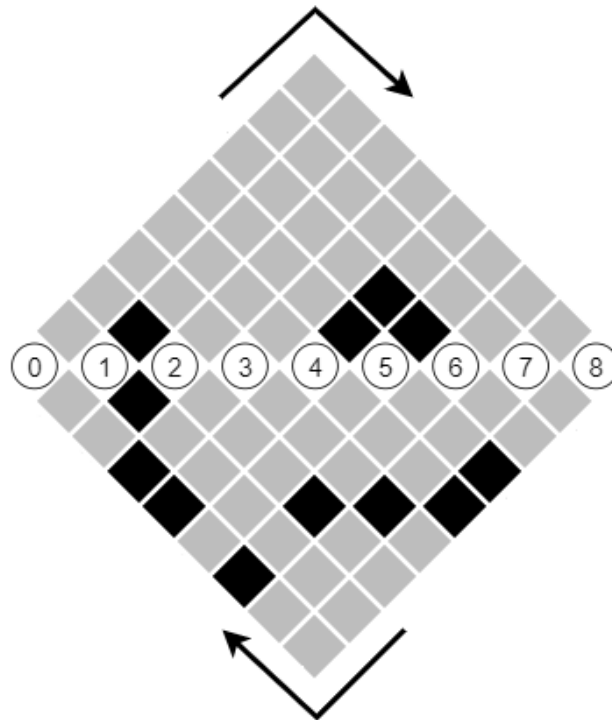


Figure 3.5: A directed network visualized as a Horizontal Cross-Diagram.

The traditional matrix is rotated 45 degrees counterclockwise. By rotating the matrix -45 degrees, the diagonal entries are now used as a central horizontal axis to represent the nodes. To emphasize this change, the diagonal entries of the matrix are encoded with circular nodes. Now, a network is represented with a diamond-shaped grid rather than a square matrix. The grid cells are used to depict the connections between the nodes. Suppose that node X is always to the left node Y on the central horizontal axis, then, the upper half of the grid depicts all outgoing links from node X to node Y, while the lower half of the grid depicts all incoming links to node X from node Y. In other words, the direction of the links is clockwise, as indicated by the arrows on both sides of the visualization in Figure 3.5.

The term “Cross-Diagram” results from the visual perception of the links corresponding to each node. Each node has a corresponding “cross” depicting its outgoing and incoming links. Figure 3.6 depicts a “cross” corresponding to a particular node.

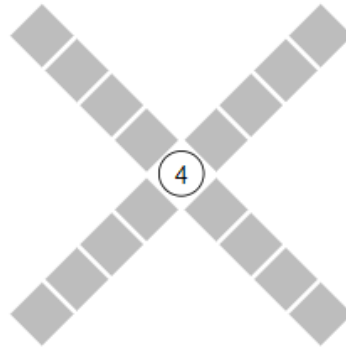


Figure 3.6: An illustration of the “cross” corresponding to node “4” in a Horizontal Cross-Diagram.

The “cross” corresponding to node “4” has four arms. The upper-right and lower-left arms depict the outgoing links of node “4”, while the upper-left and lower-right arms depict the incoming links of node “4”. Obviously, not all “crosses” are “even-sized”. Also, the “crosses” corresponding to the first and last node have two arms, such as node “0” and node “8”.

With the rotation of the matrix, orientating in the matrix happens more conveniently, because users have to move back and forth only on the central horizontal axis that depicts the nodes.

3.4 Single Curved Polyline Adjacency Matrix (SCPAM)

The *Single Curved Polyline Adjacency Matrix* (SCPAM) is proposed as a solution to the problem of the non-intuitive visual encoding of links in an AM. Figure 3.7 shows a small, directed network visualized as a SCPAM.

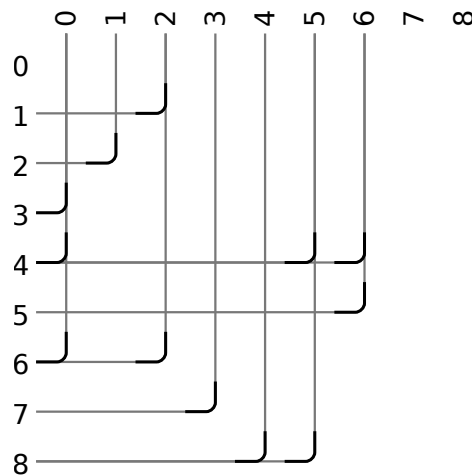


Figure 3.7: A directed network visualized as a Single Curved Polyline Adjacency Matrix.

The nodes are depicted on both the X-axis and the Y-axis, just as in the AM. Links are encoded with curved lines (polylines) connecting the nodes directly (see Figure 1.1). Sansen et al. [SBPP15] proposed a similar visualization, called *Encoding 4*. In the SCPAM, however, the distinction between the orthogonal bend of a polyline and other polylines crossing is not only addressed with curves, but also with coloring the orthogonal bend in a different color. Thus, the difference between the lines becomes more apparent. Furthermore, in contrast to *Encoding 4*, in the SCPAM, the polylines connect their respective endpoints directly, to avert unnecessary “visualization junk”.

Encoding the links with polylines rather than squares provides a more intuitive way of depicting connections between nodes because a path from one node to another can be easily traced by simply following the corresponding links in the SCPAM.

3.5 Arc Diagram (ArcD)

The *Arc Diagram* (ArcD) is proposed as a solution to the problem of laborious orientation in an AM as well as a solution to the problem of the non-intuitive visual encoding of links in an AM. The ArcD was first introduced in [Wat02]. Figure 3.8 shows a small, directed network visualized as an ArcD.

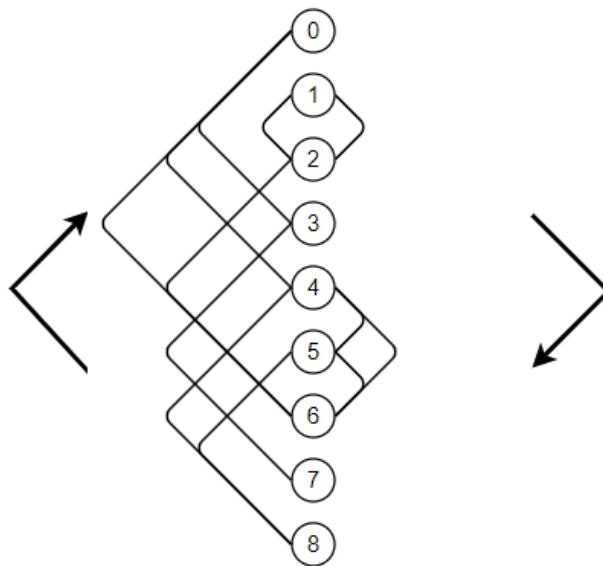


Figure 3.8: A directed network visualized as an Arc Diagram.

All nodes are positioned on a vertical axis at the center of the diagram. Links are encoded with arcs connecting the nodes directly. Suppose that node X is always above node Y on the central vertical axis, then, the right half of the visualization depicts all outgoing links from node X to node Y , while the left half of the visualization depicts all incoming links from node Y to node X . In other words, the direction of the links is clockwise, as indicated by the arrows on both sides of the visualization in Figure 3.8.

Orientating in the visualization happens more conveniently because users have to move back and forth only on the central vertical axis that depicts the nodes. Encoding the links with arcs rather than squares provides a more intuitive way of depicting connections between nodes because a path from one node to another can be easily traced by simply following the corresponding links in the ArcM.

Despite its intuitiveness, the ArcD suffers from the problem of link-crossings. Thus, the ArcD becomes cluttered quickly when the underlying network increases in density.

3.6 Arc Matrix (ArcM)

The *Arc Matrix* (ArcM) is proposed as a solution to the problem of laborious orientation in an AM as well as a solution to the problem of the non-intuitive visual encoding of links in an AM. Figure 3.9 shows a small, directed network visualized as an ArcM.

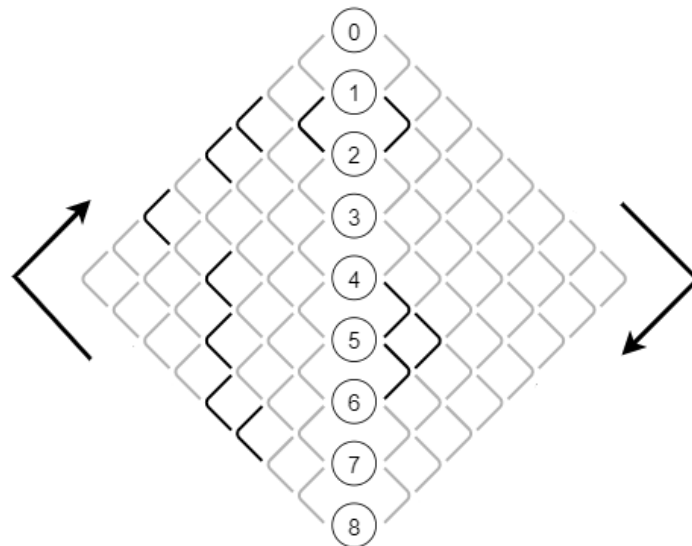


Figure 3.9: A directed network visualized as an Arc Matrix.

The traditional matrix is rotated 45 degrees clockwise. By rotating the matrix 45 degrees, the diagonal entries are now used as a central vertical axis to represent the nodes. To emphasize this change, the diagonal entries of the matrix are encoded with circular nodes. Therefore, a network is now represented with a diamond-shaped grid rather than a square matrix. The grid cells are used to depict the connections between the nodes. The links are encoded with arcs. However, only the bends of the arcs (hereafter referred to as “arc bends”) are rendered to eliminate link-crossings, and thus, visual clutter altogether. Suppose that node X is always above node Y on the central vertical axis, then, the right half of the grid depicts all outgoing links from node X to node Y, while the left half of the grid depicts all incoming links from node Y to node X. In other words, the direction of the links is clockwise, as indicated by the arrows on both sides of the visualization in Figure 3.9.

With the rotation of the matrix, orientating in the matrix happens more conveniently because users have to move back and forth only on the central horizontal axis that depicts the nodes. Encoding the links with arc bends rather than squares provides a more intuitive way of depicting connections between nodes because a path from one node to another can be easily traced by following the corresponding links in the ArcM. Following a link can be done by simply visually extending the link in both directions as depicted in Figure 3.10.

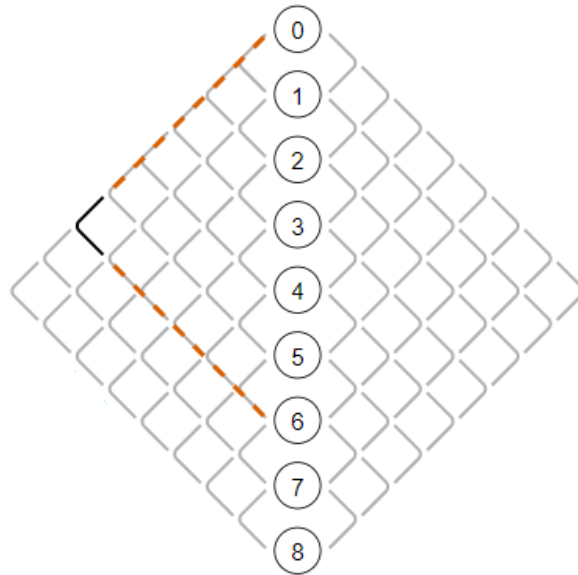


Figure 3.10: Extending a link between two nodes in an Arc Matrix.

3.7 Variants Considered for the Study

Figure 3.11 shows all variants proposed in this work, including the AM and NLD, visualizing the same network. Note that node labels have been added to each side of the VCD, HCD, and ArcM, respectively, to provide a faster lookup of nodes.

Due to budget limitations and to set the scope of the study, two of the proposed variants of AM are considered for comparison with the AM. The first variant considered for the study is the VCD. By comparing the VCD with the AM, it can be investigated how the rotation of the matrix influences the user performance at path-related tasks. It is noteworthy that the study includes a task concerning the detection of mirror symmetry (see Section 4.1). Therefore, the VCD is preferred over the HCD, because it has been proven that mirror symmetry is detected faster if the axis is vertically aligned rather than horizontally [BR79; Wag95]. The second variant considered for the study is the ArcM. By comparing the ArcM with the AM, it can be investigated how the rotation of the matrix and the visual encoding of links influences the user performance at path-related tasks. Since the VCD and ArcM only differ in the visual encoding of the links, it can be investigated how the visual encoding of links directly influences the user performance at path-related tasks.

3 Variants of Adjacency Matrix

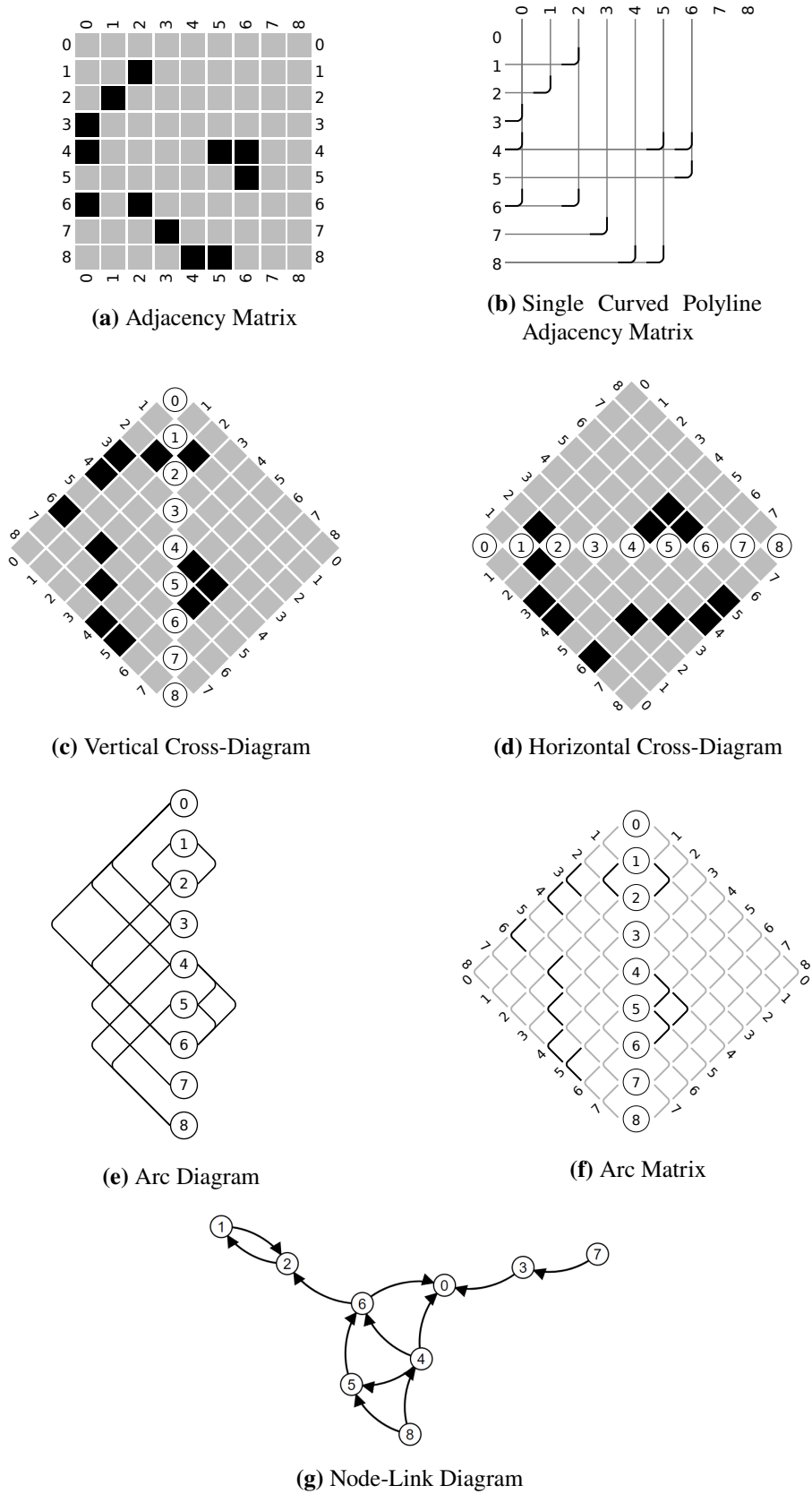


Figure 3.11: A directed network visualized using different techniques.

4 Study Design

In this chapter, the development and conduction of the study are detailed. The study was designed using jsPsych¹, a JavaScript library for running behavioral experiments in a web browser. First, Section 4.1 covers the network analysis tasks and the corresponding hypotheses. Next, Sections 4.2 and 4.3 cover the independent and dependent variables considered for the study, respectively. Section 4.4 covers confounding factors that influence both the independent variables and the dependent variables. Lastly, in Sections 4.5 and 4.6, the design and conduction of the study are described, respectively.

4.1 Tasks and Hypotheses

Table 4.1 lists the network analysis tasks that are considered for the study. Table 4.2 lists the hypotheses for how the three visualizations AM, VCD, and ArcM will compare with each other based on the user performance at the respective tasks. The user performance is measured by the accuracy (i.e. number of correct answers on average) users achieve in a task and the time users take to complete a task.

In the following, the reasoning for the hypotheses is provided for each task, respectively.

Count Neighbors (CN). Across all networks, a significantly shorter answer time will be recorded on the AM than on any other visualization. That is because the procedure of counting the neighbors of node X is shorter on the AM than on the other visualizations. On the AM, the neighbors of node X are counted in horizontal (i.e., the column of node X) and vertical (i.e., the row of node X) directions, respectively. In contrast, on the VCD and ArcM, the neighbors of node X are counted in multiple diagonal directions, which will be time-consuming and challenging to do. A higher accuracy will be achieved on the AM than on the VCD and ArcM, respectively, across all networks. That is because scanning in a diagonal direction is more challenging than scanning in a horizontal or vertical direction. Because of the rotation of the matrix, the VCD and ArcM are larger than the AM in layout size across all networks. Thus, in large networks, vertical scrolling might be required when counting the neighbors of node X, which can lead to mistakes in counting, if users forget the position they were at before scrolling up or down. Vertical scrolling additionally leads to a longer answer time. No significant difference in accuracy will be recorded on the VCD and ArcM, respectively, across all networks. However, across dense networks, a significantly shorter answer time will be recorded on the ArcM than on the VCD. That is because the visual encoding of links with arc bends is more intuitive, and therefore, links can be followed more conveniently on the ArcM than on the VCD.

¹<https://www.jspsych.org/>

Identify Link Endpoints (ILE). No significant difference in accuracy will be recorded on any visualization across all networks. However, a significantly shorter answer time will be recorded on the VCD as well as on the ArcM than on the AM across all networks. The larger the network, the more visible will be the difference in answer time on the respective visualizations. If a link is located nearby the borders of the visualization, then the node labels depicted on each side of the VCD as well as ArcM can be used to identify the endpoints to keep the distance to be traveled from the link to both its endpoints to a minimum. Similarly, the node labels depicted on the X-axis and Y-axis are used on the AM to identify the endpoints. However, the closer the link is located to the center of the AM, the greater becomes the distance to be traveled from the link to both its endpoints, because both endpoints are positioned on the X-axis and Y-axis, respectively. In contrast, on the VCD and ArcM, the distance to be traveled from the link to both its endpoints is greatly reduced, because all nodes are encoded on the central vertical axis. No significant difference in accuracy will be recorded on the VCD and ArcM, respectively, across all networks. However, across dense networks, a significantly shorter answer time will be recorded on the ArcM than on the VCD. That is because the visual encoding of links with arc bends is more intuitive, and therefore, links can be followed more conveniently on the ArcM than on the VCD.

Bidirectional Linkage (BL). A significantly higher accuracy, as well as a significantly shorter answer time, will be recorded on the VCD as well as on the ArcM than on the AM across all networks. That is because on the VCD as well as on the ArcM, a link is mirrored on the central vertical axis. It has been proven that mirror symmetry is detected faster and more accurately if the axis is vertically aligned [BR79; Wag95]. No significant difference in accuracy will be recorded on the VCD and ArcM, respectively, across all networks. However, across dense networks, a significantly shorter answer time will be recorded on the ArcM than on the VCD. That is because links encoded with arc bends take lesser screen space away than squares so that areas with a high concentration of links do not appear crowded in the ArcM. A crowded area leads to a longer answer time as users have to inspect that area thoroughly if the link is located in the area.

Direct Connection (DC). No significant difference in accuracy will be recorded on any visualization across all networks. However, a significantly shorter answer time will be recorded on the VCD as well as on the ArcM than on the AM across all networks. The larger the network, the more visible will be the difference in answer time on the respective visualizations. If the likely connection, between nodes X and Y, is located nearby the borders of the visualization, then the node labels depicted on each side of the VCD as well as ArcM can be used to keep the distances to be traveled from nodes X and Y to the position of their likely connection to a minimum. Similarly, the node labels depicted on the X-axis and Y-axis are used on the AM to locate the likely connection. However, the closer the likely connection is located to the center of the AM, the greater becomes the distance to be traveled from nodes X and Y to the position of their likely connection, because both nodes are positioned on the X-axis and Y-axis, respectively. In contrast, on the VCD and ArcM, the distance to be traveled from nodes X and Y to the position of their likely connection is greatly reduced, because all nodes are encoded on the central vertical axis. No significant difference in accuracy will be recorded on the VCD and ArcM across all networks. However, across dense networks, a significantly shorter answer time will be recorded on the ArcM than on the VCD. That is because the visual encoding of links with arc bends is more intuitive, and therefore, links can be followed more conveniently on the ArcM than on the VCD.

Find Path (FP). A significantly higher accuracy, as well as a significantly shorter answer time, will be recorded on the VCD as well as on the ArcM than on the AM across all networks. The first reason is that users have to continuously go back and forth between the rows and columns on the AM, which can be especially difficult on large networks. The second reason is that a path can not be easily traced by following the corresponding links on the AM because encoding the links with squares is not an intuitive way of depicting connections between nodes. In contrast, following paths on the VCD as well as on the ArcM does not require the user to jump back and forth between rows and columns, because in both visualizations the nodes are encoded on the central vertical axis, which enables a faster lookup of paths. A significantly higher accuracy, as well as a significantly shorter answer time, will be recorded on the ArcM than on the VCD across all networks. That is because a path can not be easily traced by following the corresponding links on the VCD. After all, encoding links with squares is not an intuitive way of depicting connections between nodes. In contrast, the arc bends in the ArcM can be extended, so that a path can be traced easily. Thus, users will not rely on the “trial and error” technique, which is not only time-consuming but also leads to quick frustration so that users quit the task midway.

Task	Details
<p>Count Neighbors (CN)</p>	<p><i>Description:</i></p> <ul style="list-style-type: none"> – Count the number of neighbors of node X – Node Y is a neighbor of node X if node X has an outgoing link to node Y or an incoming link from node Y – If node X has an incoming link from node Y as well as an outgoing link to node Y, then node Y is counted twice – Node X is highlighted in a different color <p><i>Answer Type:</i></p> <ul style="list-style-type: none"> – Number of total count
<p>Identify Link Endpoints (ILE)</p>	<p><i>Description:</i></p> <ul style="list-style-type: none"> – Identify the source node X and the target node Y corresponding to an existing link – The link is highlighted in a different color <p><i>Answer Type:</i></p> <ul style="list-style-type: none"> – The source node X and the target node Y
<p>Bidirectional Linkage (BL)</p>	<p><i>Description:</i></p> <ul style="list-style-type: none"> – Check, if the connection between two nodes is bidirectional or not – First, the network is shown for ten seconds only – The link in one direction is highlighted – Mirror the highlighted link into the other half of the diagram to determine if it exists in the opposite direction or not – After the 10 seconds pass, the network disappears – If check is completed before the ten seconds pass, it is possible to skip to the page where answer is logged in <p><i>Answer Type:</i></p> <ul style="list-style-type: none"> – “Yes”, “No” or “I don’t know”
<p>Direct Connection (DC)</p>	<p><i>Description:</i></p> <ul style="list-style-type: none"> – Check, if there exists a direct link going out from node X to node Y – Nodes X and Y are highlighted in different colors <p><i>Answer Type:</i></p> <ul style="list-style-type: none"> – “Yes” or “No”
<p>Find Path (FP)</p>	<p><i>Description:</i></p> <ul style="list-style-type: none"> – Check, if there exists a path from a source node X to a target node Y through one intermediate node R – The source node X and the target node Y are highlighted in different colors <p><i>Answer Type:</i></p> <ul style="list-style-type: none"> – In case such path does not exist → “No” – In case such path exists → Intermediate node R – If multiple such paths exist, any of those paths is considered as a correct answer, i.e., any of those intermediate nodes is considered as a correct answer

Table 4.1: List of network analysis tasks considered for the study.

Task	Hypothesis
Count Neighbors (CN)	<p><i>Accuracy:</i></p> <ul style="list-style-type: none"> – Higher on AM than on VCD and ArcM, respectively, across all networks – Similar on VCD and ArcM, respectively, across all networks <p><i>Answer Time:</i></p> <ul style="list-style-type: none"> – Shorter on AM than on VCD and ArcM, respectively, across all networks – Shorter on ArcM than on VCD across dense networks
Identify Link Endpoints (ILE)	<p><i>Accuracy:</i></p> <ul style="list-style-type: none"> – Similar on all visualizations across all networks <p><i>Answer Time:</i></p> <ul style="list-style-type: none"> – Shorter on VCD and ArcM, respectively, than on AM across all networks – Shorter on ArcM than on VCD across dense networks
Bidirectional Linkage (BL)	<p><i>Accuracy:</i></p> <ul style="list-style-type: none"> – Higher on VCD and ArcM, respectively, than on AM across all networks – Similar on VCD and ArcM, respectively, across all networks <p><i>Answer Time:</i></p> <ul style="list-style-type: none"> – Shorter on VCD and ArcM, respectively, than on AM across all networks – Shorter on ArcM than on VCD across dense networks
Direct Connection (DC)	<p><i>Accuracy:</i></p> <ul style="list-style-type: none"> – Similar on all visualizations across all networks <p><i>Answer Time:</i></p> <ul style="list-style-type: none"> – Shorter on VCD and ArcM, respectively, than on AM across all networks – Shorter on ArcM than on VCD across dense networks
Find Path (FP)	<p><i>Accuracy:</i></p> <ul style="list-style-type: none"> – Higher on VCD and ArcM, respectively, than on AM across all networks – Higher on ArcM than on VCD across all networks <p><i>Answer Time:</i></p> <ul style="list-style-type: none"> – Shorter on VCD and ArcM, respectively, than on AM across all networks – Shorter on ArcM than on VCD across all networks

Table 4.2: List of hypotheses for how the visualizations AM, VCD, and ArcM will compare with each other for each task, split by accuracy and answer time

4.2 Independent Variables

Throughout the study, we investigate the influence of three independent variables on user performance. These variables are:

1. Visualization type (AM, VCD, and ArcM)
2. Network size

3. Network density

The networks vary in three different sizes with two different densities for each size, that is to say, a total of six different networks (see Table 4.3).

Size\Density	0.1	0.2
20	Network 1	Network 2
40	Network 3	Network 4
80	Network 5	Network 6

Table 4.3: The six networks of varying complexity used for the study.

The density d in a directed network is defined as follows:

$$(4.1) \quad d = \frac{m}{n(n-1)}$$

Table 4.3 indicates that each task consists of six trials with each trial differing in size and density of the network. For simplicity, nodes in a network are labeled numerically in sequential order. The networks are not only directed but also follow the evolution patterns and characteristics that are observed in real-world networks. To synthetically generate directed networks that capture the properties of real-world networks, different algorithmic approaches are applied (see Chapter 5 for more details).

4.3 Dependent Variables

Throughout the study, the following dependent variables are measured:

- Accuracy (i.e., the number of correct answers on average)
- Answer time
- Perceived difficulty of task

Additionally, the time spent on every page of the study is measured to identify participants that rush through the experiment.

4.4 Confounding Factors

It is of great importance to consider and account for the confounding variables to ensure that the results are valid. Among other things, it is ensured that participants do not belong to a specific age group or gender so that the results can be generalized to all age groups and genders. In the study, the following confounding variables are recorded:

- Demographic data
 - Age

- Gender
 - Education
 - Level of experience with network visualization
 - Level of knowledge regarding network analysis
 - Familiarity with the term “Adjacency Matrix”
- Data concerning any vision deficiency participants potentially suffer from

4.5 Experimental Design

The study classifies as a between-subjects design, i.e., each participant is exposed to a single visualization (i.e., AM, VCD, or ArcM). Thus, learning effects are reduced, because participants do not perform the tasks again on a different visualization. The between-subjects design is implemented by setting the type of visualization manually before launching the study. Participants are recruited on Amazon’s Mechanical Turk (MTurk)². To take the study, participants have to fulfill the following requirements:

- At least 18 years of age
- Fluent in English, because the study is in English to make it accessible to a vast population
- Chrome or Firefox as the web browser for compatibility reasons³
- The laptop or desktop device used for the study must have a resolution size of at least 1920×1080

With the last requirement, it is ensured that all visualizations are fitted on the screen horizontally across large-sized networks. Without the last requirement in place, horizontal scrolling increases the difficulty of performing the network analysis tasks. Only if all requirements are met, participants can take the study. To reduce learning effects, the order of the tasks and the order of the trials within the same task are randomized. Moreover, in each trial, the variable in focus (i.e., a node, link, or node pair, depending on the task) is randomly selected from a pool of valid variables in real-time. The random selection of the variable in focus prevents the persistent scenario in which the variable in focus in any trial favors one visualization more than the others.

²<https://www.mturk.com/>

³<https://www.jpsych.org/overview/browser-device-support/>

4.6 Study Conduction

At the beginning of the study, participants have to read and accept the terms and conditions for the study. Then, data concerning the demographics of the participants are recorded. Additionally, data concerning any vision deficiency participants potentially suffer from are recorded. To this end, participants take a color-blindness test, since colors are applied throughout the study, for example, to highlight a node or link in a visualization. As a precautionary measure, only color-blind friendly colors are applied throughout the study. After recording a variety of data concerning the participants, they are introduced to the network theory. Afterward participants are introduced to a NLD since it is a very easy-to-understand visualization type. Next, participants are introduced to one of the visualizations (i.e., AM, VCD, or ArcM) on that they perform the tasks. Then, participants perform several network analysis tasks on that visualization. Figure 4.1 illustrates the overall structure of the study.

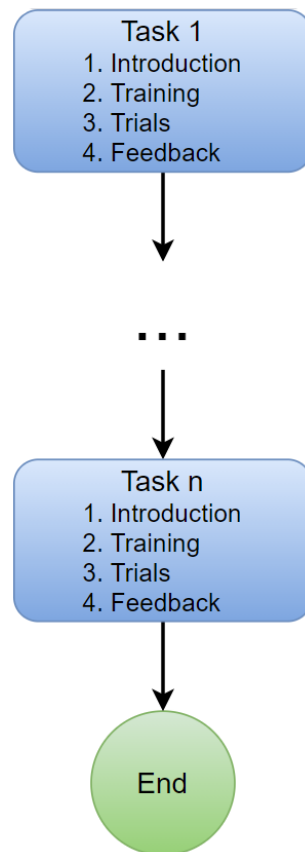


Figure 4.1: A flow chart showing how participants proceed from task to task.

First, an explanation of the task is provided. The introduction to the task also includes a small example of the task. Next, participants perform two training trials on small-sized networks, to gain an understanding of how to perform the task correctly. Then, participants perform the main trials of the task. After completing a main trial, participants are asked to rate the difficulty of the task on a Likert scale ranging from “Very Difficult” to “Very Easy”. After completing all main

trials, participants are asked to explain the approach they took to perform the task. Additionally, participants can share feedback that can concern either the task, the used visualization, or the underlying network. Once participants complete all tasks, they must submit the study to receive a randomly generated code. Participants submit the code on MTurk, to receive their payment for taking part in the study.

5 Network Data

Figure 5.1 shows a 3-step solution for synthetically generating directed networks that capture the properties of real-world networks.

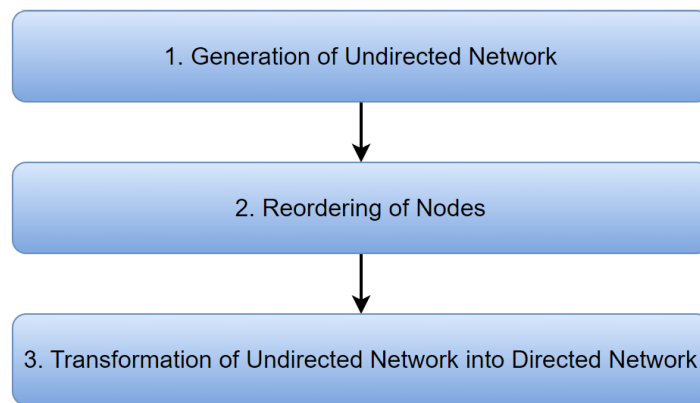


Figure 5.1: A flow chart depicting the steps for generating directed networks.

All steps are implemented using the programming language Python ¹. Step 1 is implemented using an existing algorithm called FARZ [FARZ18]. Step 2 is implemented using the Python libraries NumPy ², SciPy ³, and Matplotlib ⁴. NumPy provides a multidimensional array object, Scipy provides clustering algorithms among other things, and Matplotlib provides visualization capabilities. Step 3 is implemented using a self-developed algorithm. In the following, each step is laid out in detail.

Step 1: Generation of undirected network

Fagnan et al. [FARZ18] developed a network generator called FARZ that synthesizes the characteristics of the networks and communities that are observed in real-world networks. The name FARZ is derived from the surnames of the authors. Originally, FARZ was developed to serve as a benchmark for validating and comparing community detection methods. However, for this study, FARZ is solely used for generating networks. Algorithm 5.1 illustrates how the networks are generated.

¹<https://www.python.org/>

²<https://numpy.org/>

³<https://www.scipy.org/>

⁴<https://matplotlib.org/>

Algorithm 5.1 Generator(n, m, k)

```

1: N ← Network()
2: C ← {c1 = ∅, c2 = ∅, ... ck = ∅} // Initialize
3: for i in [1 ... n] do
4:   N.add_node(i) // Add node i
5:   assign(i, C) // Assign i to communities
6:   connect(i, G, C) // Add a link from node i
7:   for j in [2 ... m] do // Add m - 1 links
8:     j ← select(N.nodes) // Select node j from N
9:     connect(j, N, C) // Add a link from node j
10:  end for
11: end for
12: return N, C

```

The *Generator* function takes the following three arguments as input:

- n : The number of nodes
- k : The number of communities
- m : The number of links added at each step

The network is expanded one node at a time. Each node i is added to the network (see line 4 of Algorithm 5.1) and assigned to r communities (see line 5 of Algorithm 5.1). For the study, the aim is to keep the networks simple in regards to the community membership of a node. Therefore, r is set to 1, so that networks with non-overlapping communities are generated. Node i is assigned to a community u with a certain probability:

$$(5.1) \quad p(u) = \frac{|u| + \phi}{\sum_v (|v| + \phi)}$$

The denominator is a normalizing factor that sums up the sizes of all communities. By setting ϕ to 1, it is ensured that empty communities also have a chance to recruit nodes. As ϕ is increased, the distribution for sizes of communities becomes closer to uniform, resulting in equal-sized communities in the network. For the study, ϕ is set to 1, to generate networks with a power-law distribution for community sizes. This type of distribution is also observed in social networks [AB02]. After adding node i to the network and assigning it to a community u , connections are formed between nodes. However, connections are not only formed for node i (see line 6 of Algorithm 5.1) but also for a randomly selected subset of nodes that were added to the network earlier (see lines 8 and 9 of Algorithm 5.1). As links are added at each round, nodes that were added earlier benefit the most because they get more chances to get selected and form connections, which naturally enforces the power-law distribution which is observed in real-world networks [FARZ18]. Algorithm 5.2 illustrates how a connection between two nodes is formed.

Algorithm 5.2 Connect(i, N, C)

```
1: if random <  $\beta$  then // Choose a community from
2:    $c \leftarrow \text{select}(\{c, \forall c \in C \wedge i \in c\})$  // Memberships of  $i$ 
3: else
4:    $c \leftarrow \text{select}(\{c, \forall c \notin C \wedge i \in c\})$  // Other communities
5: end if // Choose a node within the selected community
6:  $j \leftarrow \text{choose}(\{j, \forall j \in c \wedge j \neq i \wedge (i, j) \notin N.\text{links}\})$ 
7:  $N.\text{add\_link}(i, j)$ 
```

The *Connect* function takes the following three arguments as input:

- i : The node in focus
- N : The undirected network
- C : The communities in the network

$\beta \in [0,1]$ is a control parameter that determines the strength of the overall community structure. As β is increased, the community structure becomes more visible. If β is set to 1, then each node forms connections within its communities. For the study, the aim is to display the community structure across all networks. Therefore, β is set to 0.8. With a probability of β , node i forms its connection within its communities (see line 2 of Algorithm 5.2), otherwise node i forms a connection with a node from another community (see line 4 of Algorithm 5.2). Once a community is selected, node i forms a connection with a node j from the selected community with a certain probability (see line 6 of Algorithm 5.2). This probability depends on two factors:

1. Number of nodes i 's and j 's common neighbors
2. Similarity of nodes i 's and j 's degrees (i.e., the connections)

The first factor leads to a high clustering coefficient (i.e., the degree to which nodes in a network tend to cluster together) which is observed in real-world networks, particularly in social networks [FARZ18; WS98]. The second factor realizes assortative mixing [FARZ18]. A network shows assortative mixing when the nodes in the network that have many connections tend to be connected to other nodes with many connections [New03]. For example, social networks are assortative [FARZ18; New03]. After a node j is chosen, a connection between the nodes i and j is formed and added to the network (see line 7 of Algorithm 5.2).

Step 2: Reordering of nodes

This step concerns the presentation of a network rather than the data itself. Nodes are reordered in such a way that clusters are displayed in a matrix-based visualization. Displaying clusters in a matrix-based visualization across all networks is important, because not only does the visibility of clusters provide a cleaner layout across the networks, but it also increases the efficiency at performing path-related tasks. For example, if a path-related task involves two nodes that belong to the same community, then searching for a path between those nodes is easier to do on a matrix-based

visualization where clusters are visible because nodes that belong to the same community are positioned closer to each other. Figure 5.2 shows the steps that are performed to display the clusters in a matrix-based visualization of a network.

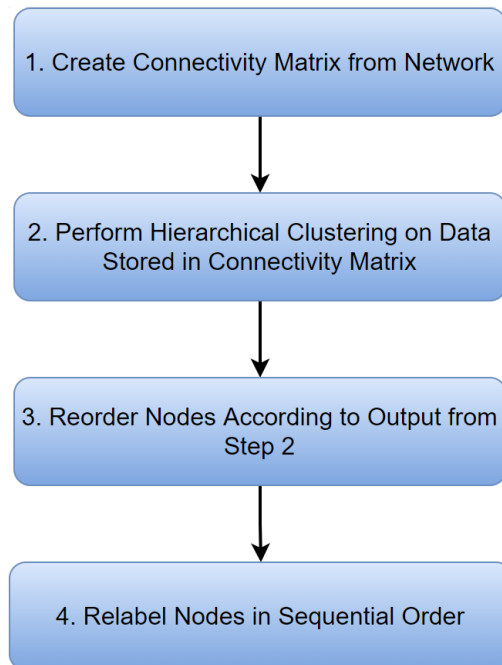


Figure 5.2: A flow chart depicting the steps for displaying the clusters in a visualization for a network.

In the first step, a connectivity matrix is created from the network. One can think of a connectivity matrix as an AM without the mapping of data variables (i.e., the nodes and links) to the respective visual variables. Instead, links are mapped to boolean values (i.e., 1s for existing links, otherwise 0s). In the next step, hierarchical clustering is performed on the data that is stored in the connectivity matrix. Hierarchical clustering is an algorithm that groups similar objects (i.e., the nodes in the network) into groups called clusters. At the end of the clustering process, each cluster is distinct from every other cluster, and the nodes within each cluster are largely similar to each other [MD18]. Hierarchical clustering is performed with a distance matrix. The connectivity matrix serves as the distance matrix because the distance between each pair of a node is either 1 (i.e., a link exists between two nodes) or 0 (i.e., a link does not exist between two nodes). The following pseudo-code illustrates the procedure of hierarchical clustering:

1. *Each node is treated as a separate cluster*
2. *Repeat the following steps until one cluster remains:*
 - 2.1. *Identify the two clusters that are closest together*
 - 2.2. *Merge the two most similar clusters*

The distance between two clusters is computed using the Euclidean distance. As for the linkage criterion, Wards' method is applied, which groups two clusters based on the sum of squared distances of each cluster [RVN20]. The hierarchical clustering outputs a dendrogram (or tree) that displays the arrangements of the clusters [MD18]. Figure 5.3 shows a dendrogram of a network of size 20.

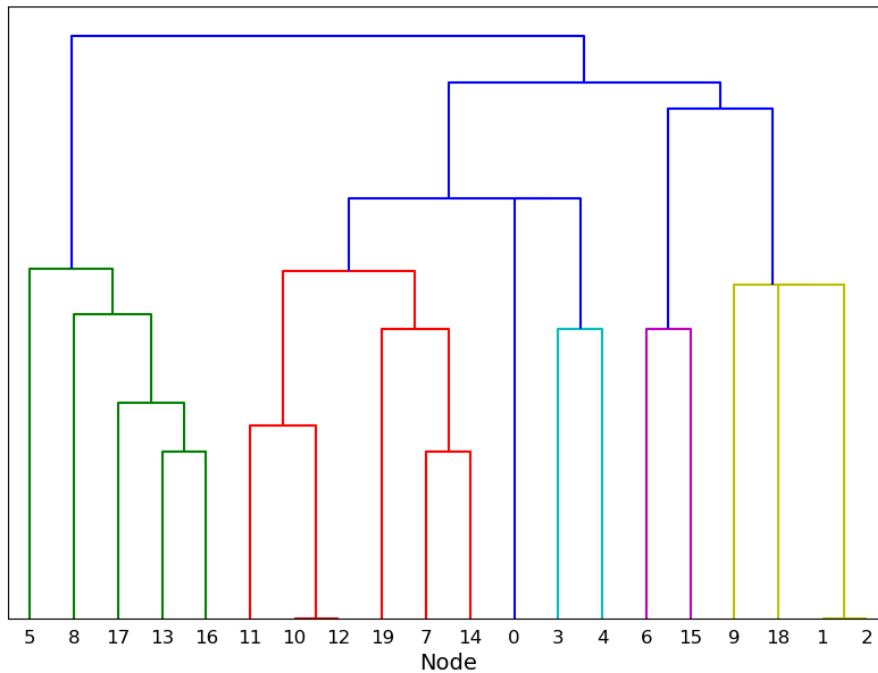
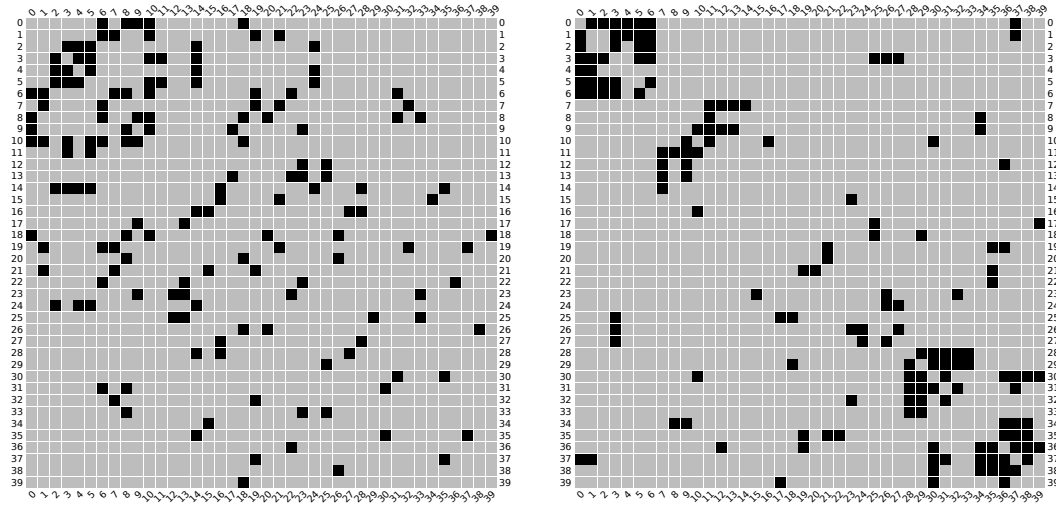


Figure 5.3: A dendrogram depicting the clusters in a network.

The dendrogram shows the relationships between the clusters. More importantly, the clustering suggests which nodes should be positioned close to each other in the matrix-based visualization. For example, according to the output in Figure 5.3, nodes “10” and “12” should be positioned close to each other in the visualization. Indeed, when looking into the connectivity matrix, both nodes are connected to the same nodes. On the other hand, nodes “5” and “2” should be positioned far away from each other. Indeed, when looking into the connectivity matrix, both nodes are connected to completely different nodes. In Step 3, the nodes are reordered according to the positions they take in the dendrogram. This means that node “5” appears at the first position in the visualization, while node “2” appears at the last position. In the last step, the nodes are relabeled in sequential order for the sake of readability. This means that node “5” is relabeled to “0”, “8” to “1”, and so on.

Figure 5.4 shows an example of how the reordering of nodes affects the overall display of the network visualized as an AM.



(a) Network before reordering of nodes

(b) Network after reordering of nodes

Figure 5.4: Illustration of the effect of a reordering process.

Step 3: Transformation of undirected network into directed network

For the transformation of an undirected network into a directed network, it suffices to “transform” the existing links in an undirected network into links that conform to the properties of a directed network. In a directed network, each link has a direction associated with it. Figure 5.5 depicts how link directions are realized in a directed network.

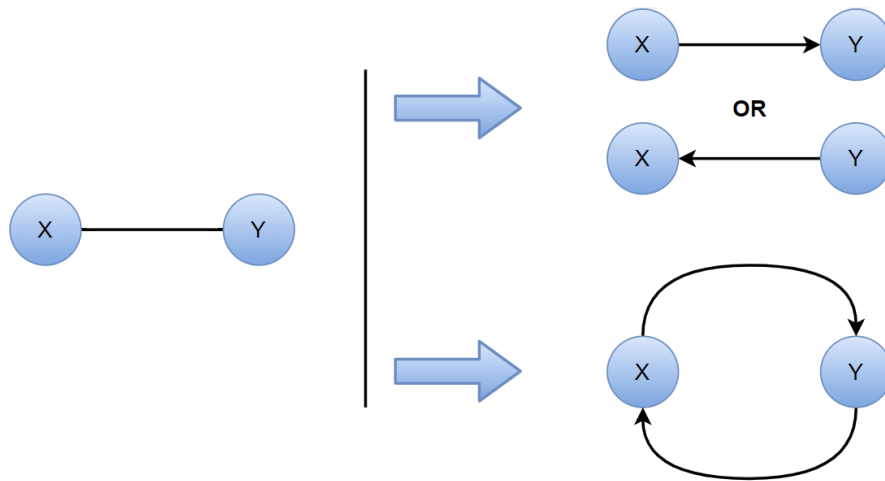


Figure 5.5: Two possible ways of depicting a link in a directed network are shown. The upper right half of the figure shows the two possible unidirectional links between two nodes, while the lower right half of the figure shows a bidirectional link between two nodes.

- Unidirectional link ($X \Rightarrow Y$): Source node X has an outgoing link to target node Y and not vice versa.

- **Bidirectional link** ($X \Leftrightarrow Y$): Source node X has an outgoing link to target node Y and vice versa.

It is important to note that a bidirectional link between two nodes is realized using two links pointing in opposite directions. Algorithm 5.3 illustrates how the existing links in an undirected network are “transformed” into either unidirectional links or bidirectional links.

Algorithm 5.3 Add_Directions_To_Links(N, s)

```

1: prev_num_of_links ← N.links.length           // Store number of links in a variable
2: for link in N.links do
3:   if random ≤  $s$  then
4:     N.add_direction(link.source, link.target) // Assign link a direction from source to target
5:     N.add_direction(link.target, link.source) // And vice versa
6:   else                                     // Assign link a direction from source to target or vice versa
7:     head ← coin_flip(0,1)
8:     if head = 0 then
9:       N.add_direction(link.source, link.target)
10:    else
11:      N.add_direction(link.target, link.source)
12:    end if
13:  end if
14:  curr_num_of_links = N.links.length         // Store current number of nodes in a variable
15:  add_missing_links(N, prev_num_of_links, curr_num_of_links)

```

The *Add_Directions_To_Links* function takes the following two arguments as input:

- N : An undirected network
- $s \in [0,1]$: A parameter for controlling the probability of a link becoming bidirectional

The closer the parameter s is set to 1, the more likely is it that an existing link becomes bidirectional. If s is set to 1, then all links become bidirectional. For the study, s is set to 0.8. In the first line of the algorithm, the number of links that exist in the undirected network is stored. For the moment, this variable is not relevant. Next, each existing link is “transformed” either into a bidirectional link with a probability of s or into a unidirectional link with a probability of $1 - s$ (see line 3). In the case that a link becomes bidirectional, two links are added between the source and target nodes, pointing in opposite directions (see lines 4-5). In the case that a link becomes unidirectional, the direction of the link is decided by a “coin flip” (see lines 7-12). After each existing link is “transformed”, the network is finally directed. The number of links that currently exist in the directed network is stored in a separate variable. Lastly, the *Add_Missing_Links* function is called with certain parameters, some of them which were previously allocated some data. This function “restores” the density of the undirected network for the directed network by adding a certain number of unidirectional links to the directed network. Algorithm 5.4 illustrates how the density is “restored”.

Algorithm 5.4 Add_Missing_Links(N , prev_num_of_links, current_num_of_links)

```
1: if curr_num_of_links < 2 · prev_num_of_links then
2:   num_of_links_to_add ← (2 · prev_num_of_links - curr_num_of_links)
3:   num_of_links_added ← 0 // A counter is set
4:   while num_of_links_added < num_of_links_to_add do
5:     source_node ← random_pick( $N$ .nodes)
6:     target_node ← random_pick( $N$ .nodes)
7:     if source_node ≠ target_node then // Avoid self-loops
8:       link ← create_link(source_node, target_node)
9:       if link in  $N$ .links then // Check whether or not new link already exists in network
10:        pass
11:       else
12:          $N$ .add_link(link) // Add new link to network
13:         num_of_links_added ← num_of_links_added + 1 // Increase counter
14:       end if
15:     else
16:       pass
17:     end if
18:   end while
19: end if
20: return  $N$ 
    =0
```

The *Add_Missing_Links* function takes the following three arguments as input:

- N : The directed network from the caller function
- *prev_num_of_links*: The number of links in the undirected network (see line 1 of Algorithm 5.3)
- *current_num_of_links*: The number of links in the directed network (see line 14 of Algorithm 5.3)

In the first line of the algorithm, the number of links in the directed network is compared with twice the number of links in the undirected network. That is because, for the same number of nodes and links, the density of a directed network is half the density of an undirected network. If the condition is met, then all links in the directed network must be bidirectional. Otherwise, the number of links that need to be added to fulfill the condition is calculated (see line 2 of Algorithm 5.4). As long as the number of links that need to be added is not reached, a source and target nodes are randomly selected from the network. If they are not identical, a link is created between the two nodes. If this link does not already exist in the network, then it is added to the network. After the missing links are added to the network, the density of the undirected network is “restored” for the directed network.

6 Results

In this chapter, the visualizations (i.e., AM, VCD, and ArcM) are evaluated based on the overall user performance at each task across all networks. The user performance is measured by the accuracy (i.e. number of correct answers on average) users achieve in a task and the time users take to complete a task. The overall accuracy is visualized using a bar chart which depicts the percentage of correct answers as a bar, for each visualization respectively. A bar also depicts the corresponding confidence interval, which is set to 95%. The individual answer times are visualized using a box plot, for each visualization respectively. A box plot depicts the distribution of the answer time. The box limits indicate the range of the central 50% of the answer times, with a horizontal line marking the median answer time.

In each section, first, the influence of the network properties, i.e., size and density, on the respective visualizations for accuracy and answer time is evaluated. Then, the visualizations are compared pairwise with each other for accuracy and answer time.

To confirm or reject a hypothesis for a task, the difference in user performance recorded on the respective visualizations has to be significant. To decide whether a difference in user performance recorded on two visualizations, respectively, is significant or not, a p-value $\in [0,1]$ is calculated using Welch's t-test (see Appendix B). Welch's t-test is a statistical method for comparing the means of two independently collected observations (i.e., accuracy or answer time) to determine if there is a significant difference between the two [DW16]. If the corresponding p-value is less than the significance level $\alpha = 0.05$, the difference is deemed significant. Also, to decide if a network property influences the performance recorded on one visualization significantly, a p-value is calculated using Welch's t-test (see Appendix A).

In total, 98 participants were recruited for the study. Some of the participants did not submit the study, while other submissions were rejected due to low-effort answers. Submissions were rated as low-effort if participants completed the study in under 10 minutes and achieved an accuracy of 30% overall tasks. Hence, 77 submissions (AM = 28, VCD = 26, ArcM = 23) were considered for the evaluation process.

The validity of the results can be confirmed, as, on one hand, each category of the respective confounding factors is not overly represented (see Appendix C), and on the other hand, no severe vision deficiencies were recorded (see Appendix D).

6.1 Count Neighbors (CN)

The task concerns counting the neighbors of a node (see Section 4.1).

Figure 6.1 shows the accuracy achieved on the respective visualizations across each network. The size of a network does not influence the accuracy on any visualization significantly (see Table A.2). The density of a network does not influence the accuracy on the AM as well as VCD significantly (see Table A.3). However, the accuracy on the ArcM decreases significantly as the network increases in density, if the network is large-sized.

Figure 6.2 shows the distribution of answer time recorded on the respective visualizations across each network. As the network increases in size, the answer time on each visualization increases significantly (see Table A.5). Network density does not influence the answer time on the VCD as well as ArcM significantly (see Table A.6). However, the answer time on the AM increases significantly as the network increases in density, if the network is medium-sized (see Table A.6).

Users achieve a higher accuracy on the AM than on any other visualization across all networks (see Figure 6.1 and Table A.1). Notably, significantly higher accuracy is achieved on the AM than on the VCD for “Network 1” (see Table B.1). No significant difference in accuracy is recorded on the VCD and ArcM, respectively, across all networks (see Table B.1).

On average, users answer faster on the AM than on any other visualization across mostly all networks (see Table A.4). In contrast, users take a longer time to answer on the ArcM than on any other visualization across all networks (see Table A.4). Notably, a significantly shorter answer time is recorded on the AM than on the ArcM across all networks, except “Network 4” (see Table B.2). Also, a significantly shorter answer time is recorded on the VCD than on the ArcM for “Network 2”, “Network 3” and “Network 4” (see Table B.2).

6.2 Identify Link Endpoints (ILE)

The task concerns identifying the two endpoints of a link (see Section 4.1).

Figure 6.3 shows the accuracy achieved on the respective visualizations across each network. Neither network size nor network density influence the accuracy on any visualization significantly (see Tables A.8 and A.9).

Figure 6.4 shows the distribution of answer time recorded on the respective visualizations across each network. As the network increases in size, the answer time on each visualization increases significantly (see Table A.11). Network density does not influence the answer time on any visualization significantly (see Table A.12).

Users achieve a higher accuracy on the AM than on any other visualization across all networks, except for “Network 6” (see Figure 6.3 and Table A.7). Notably, significantly higher accuracy is achieved on the AM than on the VCD for “Network 1” (see Table B.3). Also, a significantly higher accuracy is achieved on the AM than on the ArcM for “Network 3” (see Table B.3).

On average, users answer significantly faster on the AM than on any other visualization across all networks (see Table B.4). No significant difference is recorded in answer time on the VCD and ArcM, respectively (see Table B.4).

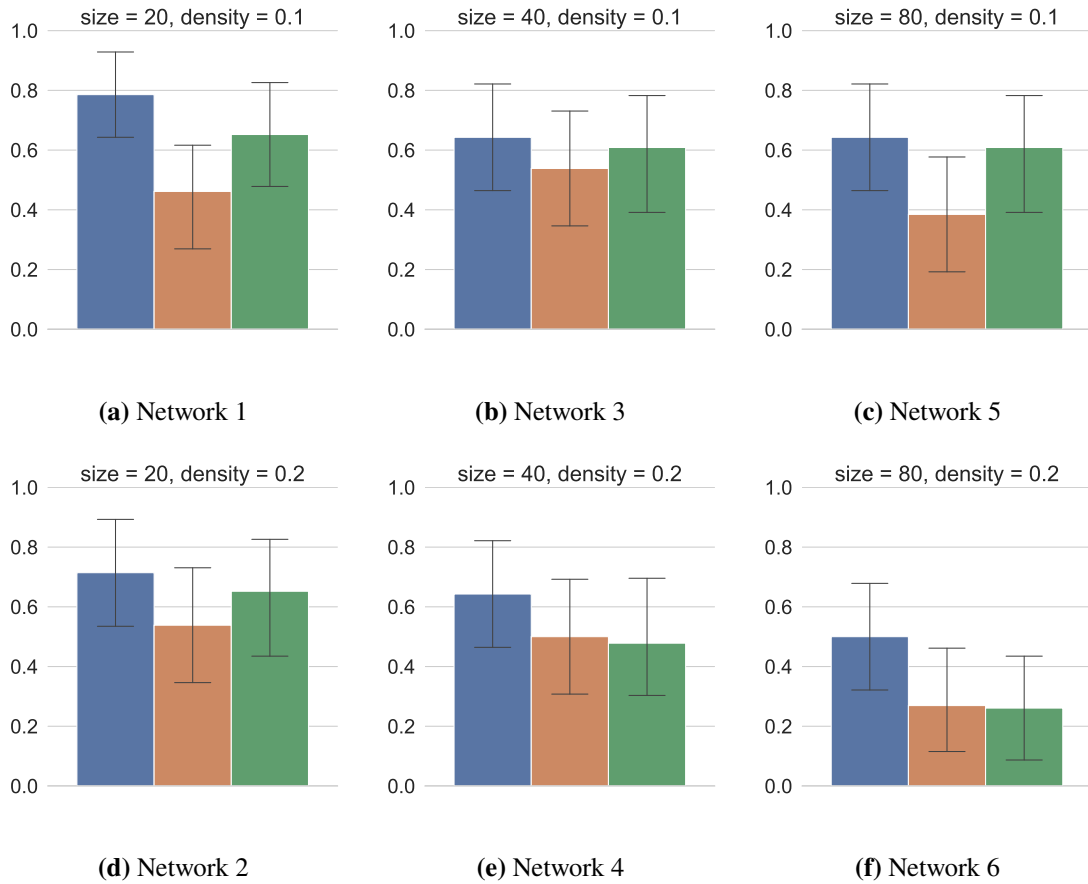


Figure 6.1: Illustration of the accuracy achieved on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for CN task. The network increases in size/density from left to right/top to bottom.

6.3 Bidirectional Linkage (BL)

This task concerns finding out whether or not a link exists in the opposite direction in the visualization (see Section 4.1).

Figure 6.5 shows the accuracy achieved on the respective visualizations across each network. Network size influences the answer time on the AM moderately (see Table A.14). It does not influence the accuracy on the VCD significantly (see Table A.14). An increase in network size leads to a significant decrease in accuracy on the ArcM, if the network becomes large-sized (see Table A.14). Network density does not influence the accuracy on the AM as well as ArcM significantly (see Table A.15). However, as the network increases in size, the accuracy on the VCD increases significantly, if the network is large-sized (see Table A.15).

Figure 6.6 shows the distribution of answer time recorded on the respective visualizations across each network. Network size influences the answer time on the AM moderately (see Table A.17). It does not influence the answer time on the VCD as well as ArcM significantly (see Table A.17).

6 Results

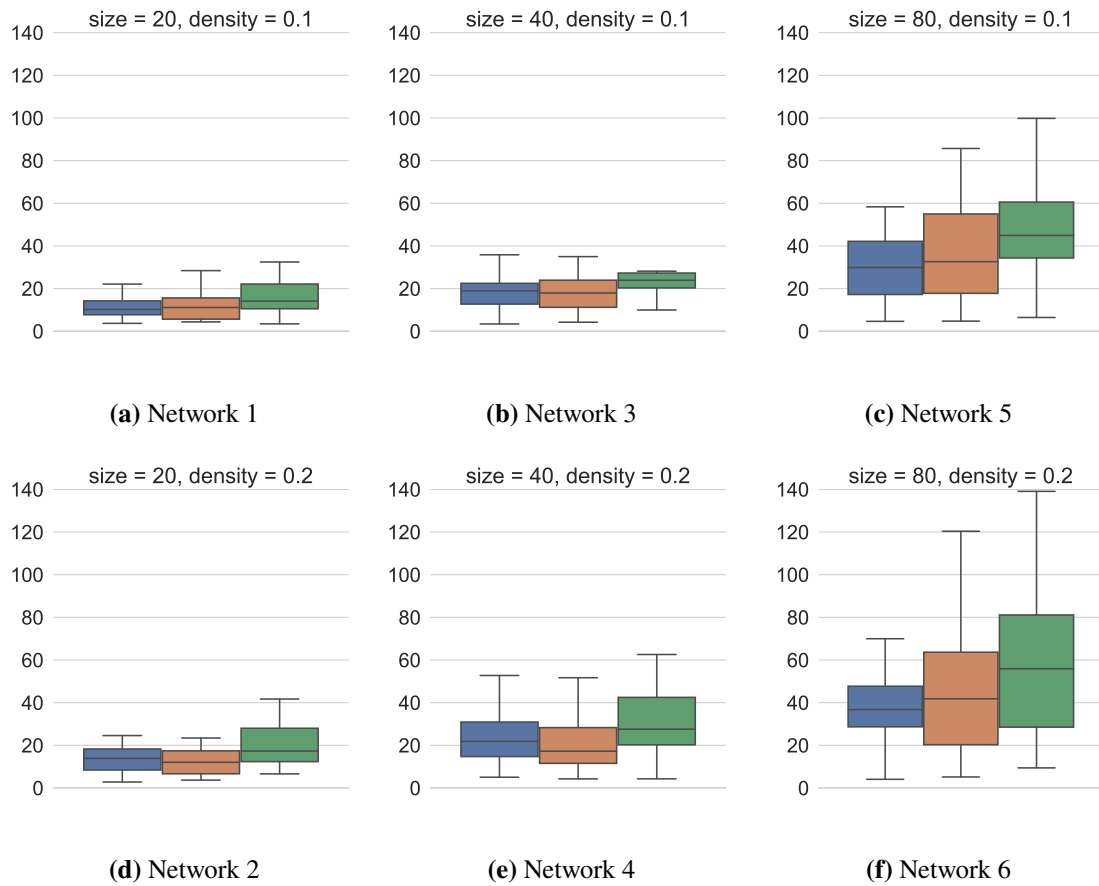


Figure 6.2: Illustration of the distribution of answer time recorded on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for CN task. The network increases in size/density from left to right/top to bottom.

An increase in network density leads to an increase in answer time on the AM, if the network is large-sized (see Table A.18). Network density does not influence the answer time on the VCD as well as ArcM significantly (see Table A.18).

Users achieve higher accuracy on the ArcM than on any other visualization across all networks, except for “Network 6” (see Figure 6.5 and and Table A.13). Notably, significantly higher accuracy is achieved on the ArcM than on the AM for “Network 3” and “Network 4” (see Figure B.5). For “Network 6”, users achieve higher accuracy on the VCD than on any other visualization. Notably, significantly higher accuracy is achieved on the VCD than on the AM for “Network 6” (see Figure B.5).

On average, the answer time on each visualization is similar across all networks, except for “Network 6”, where a significantly shorter answer time is recorded on the ArcM than on the AM (see Figure B.6).

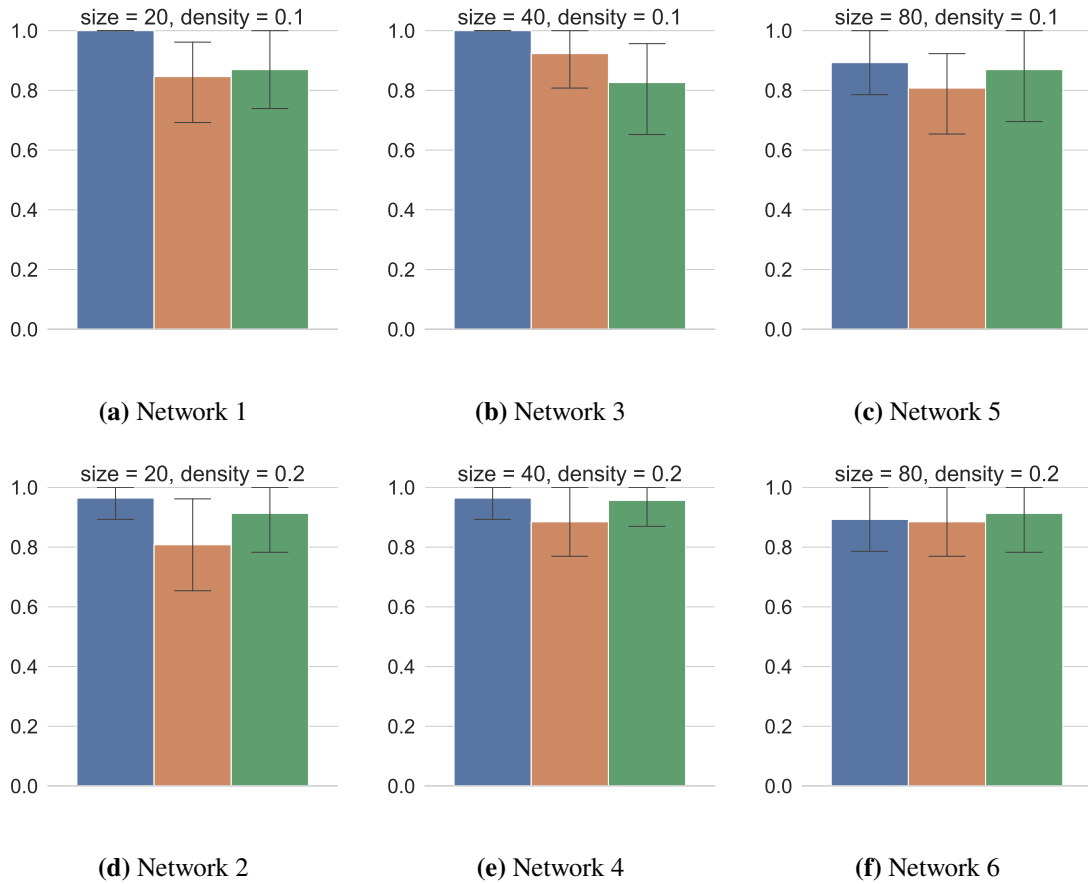


Figure 6.3: Illustration of the accuracy achieved on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for ILE task. The network increases in size/density from left to right/top to bottom.

6.4 Direct Connection (DC)

The task concerns finding out whether nodes are directly connected with each other or not (see Section 4.1).

Figure 6.7 shows the accuracy achieved on the respective visualizations across each network. Neither network size nor network density influence the accuracy on any visualization significantly (see Tables A.20 and A.21).

Figure 6.8 shows the distribution of answer time recorded on the respective visualizations across each network. An increase in network size leads to a significant increase in answer time on the AM (see Table A.23). It moderately influences the answer time on the VCD as well as ArcM (see Table A.23). Network density does not influence the answer time on the AM as well as VCD significantly (see Table A.24). However, an increase in network density leads to a significant increase in answer time on the ArcM, if the network is large-sized (see Table A.24).

No significant difference in accuracy is recorded on the respective visualizations across all networks (see Table B.7).

6 Results

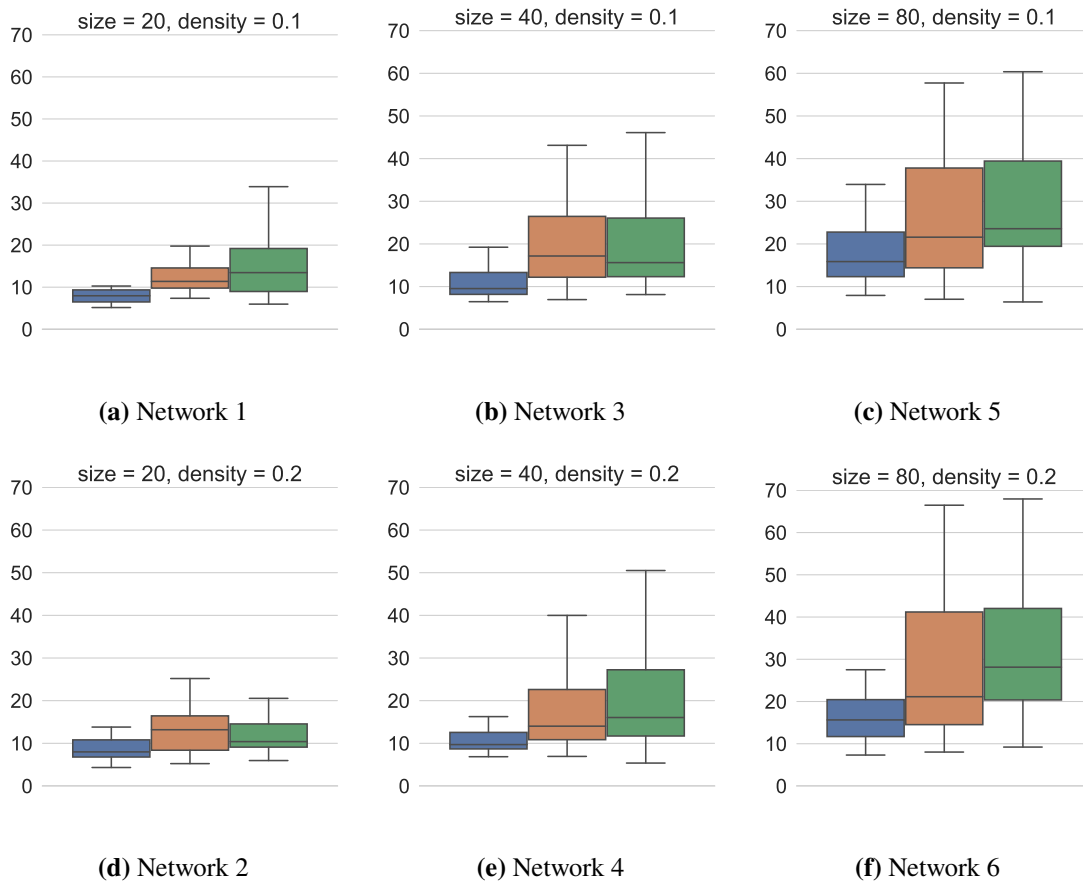


Figure 6.4: Illustration of the distribution of answer time recorded on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for ILE task. The network increases in size/density from left to right/top to bottom.

On average, users answer faster on the AM than on any other visualization across almost all networks (see Table A.22). Notably, a significantly shorter answer time is recorded on the AM than on the VCD for “Network 1” (see Table B.8). Also, a significantly shorter answer time is recorded on the AM than on the ArcM for more than half of the networks (see Table B.8). For “Network 2”, users take a significantly longer time to answer on the ArcM than on any other visualization (see Table B.8). However, a significantly shorter answer time is recorded on the ArcM than on the VCD for “Network 5” (see Table B.8).

6.5 Find Path (FP)

The task concerns finding out whether a path between two nodes through one intermediate node exists or not (see Section 4.1).

Figure 6.9 shows the accuracy achieved on the respective visualizations across each network. Neither network size nor network density influence the accuracy on any visualization significantly (see Tables A.26 and A.27).

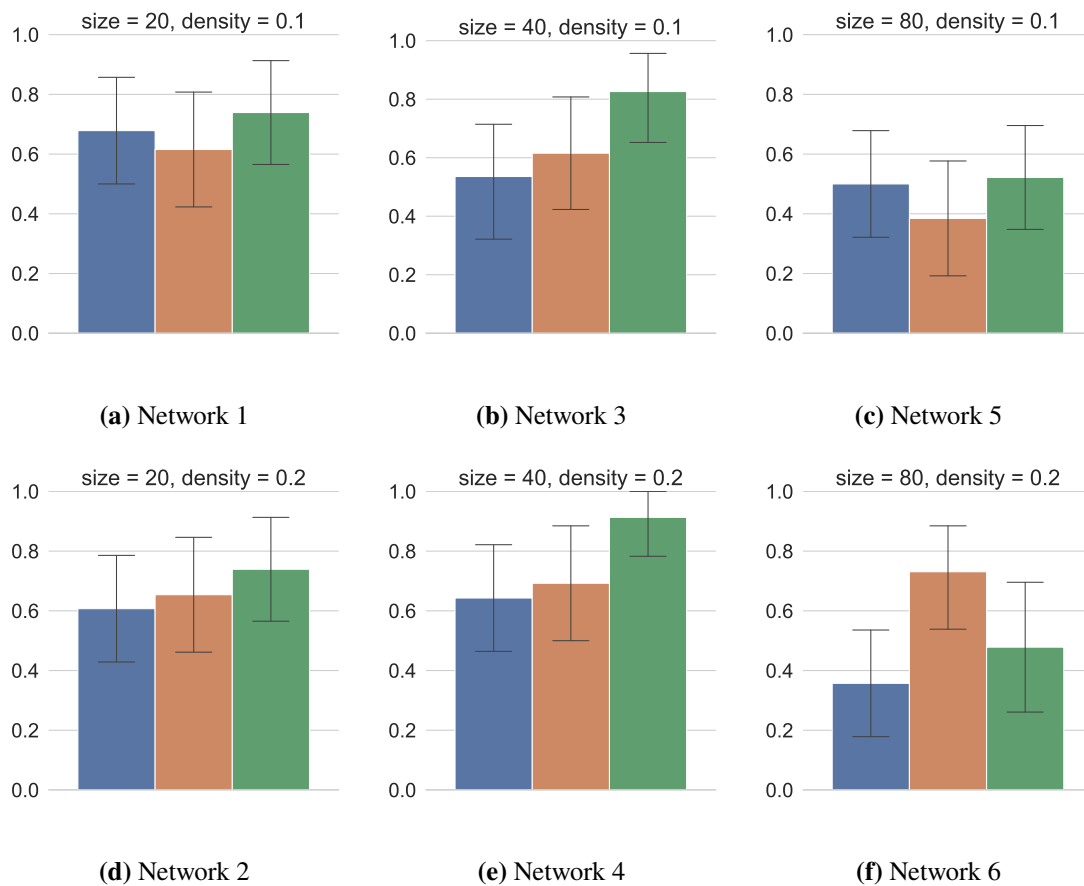


Figure 6.5: Illustration of the accuracy achieved on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for BL task. The network increases in size/density from left to right/top to bottom.

Figure 6.10 shows the distribution of answer time recorded on the respective visualizations across each network. An increase in network size leads to a significant increase in answer time on the AM, with one exception (see Table A.29). Network size moderately influences the answer time on the VCD (see Table A.29). An increase in network size leads to a significant increase in answer time on the ArcM, if the network becomes middle-sized (see Table A.29). An increase in network density leads to an increase in answer time on the AM, if the network is small-sized (see Table A.30). Network density does not influence the answer time on the VCD as well as ArcM significantly (see Table A.30).

Even though the difference in accuracy recorded on the respective visualizations is visible across almost all networks (see Figure 6.9), none of them is deemed significant (see Table B.9).

On average, users take longer time to answer on the ArcM than on any other visualization across almost all networks (see Table A.28). Users take significantly longer time to answer on the ArcM than on any other visualization across “Network 3”, and “Network 4” (see Table B.10). Also, a significantly longer answer time is recorded on the ArcM than on the AM for “Network 1”, too (see

6 Results

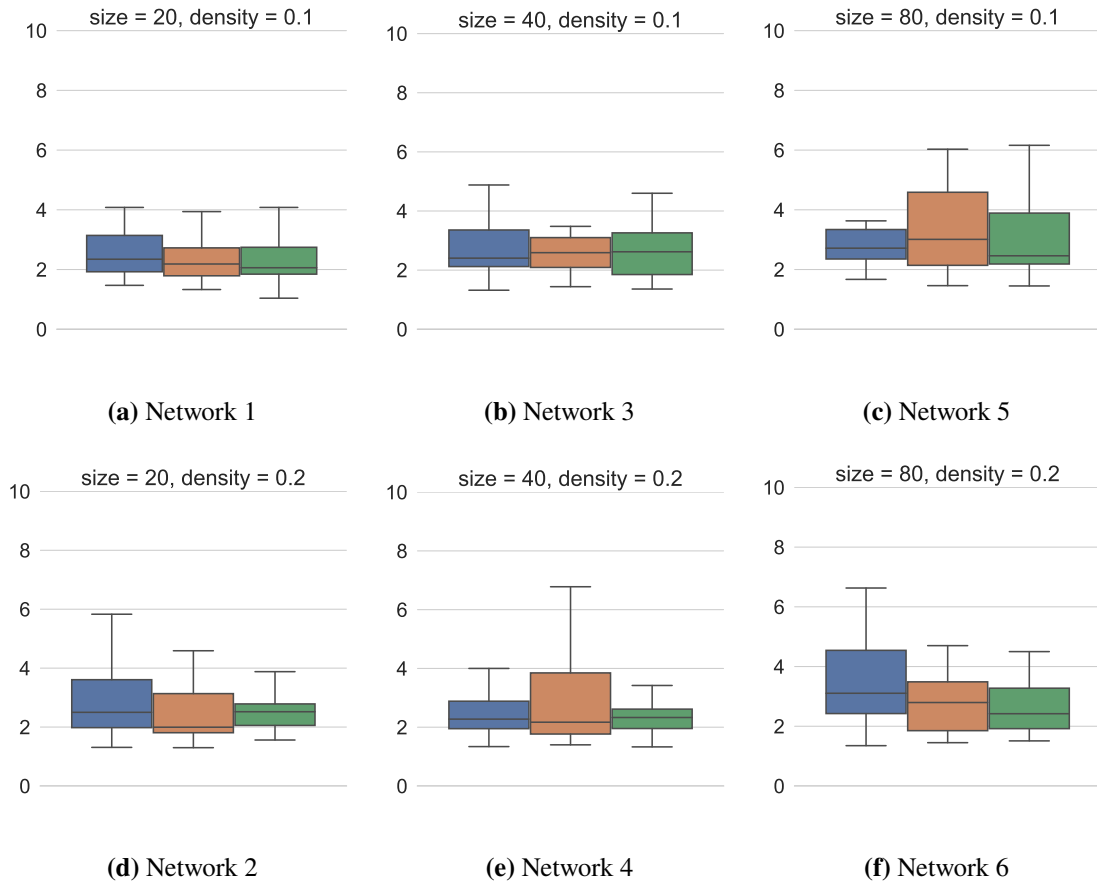


Figure 6.6: Illustration of the distribution of answer time recorded on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for BL task. The network increases in size/density from left to right/top to bottom.

Table B.10). Furthermore, a significantly longer answer time is recorded on the ArcM than on the VCD for “Network 3”, too (see Table B.10). A significantly shorter answer time is recorded on the AM than on the VCD for “Network 1” (see Table B.10).

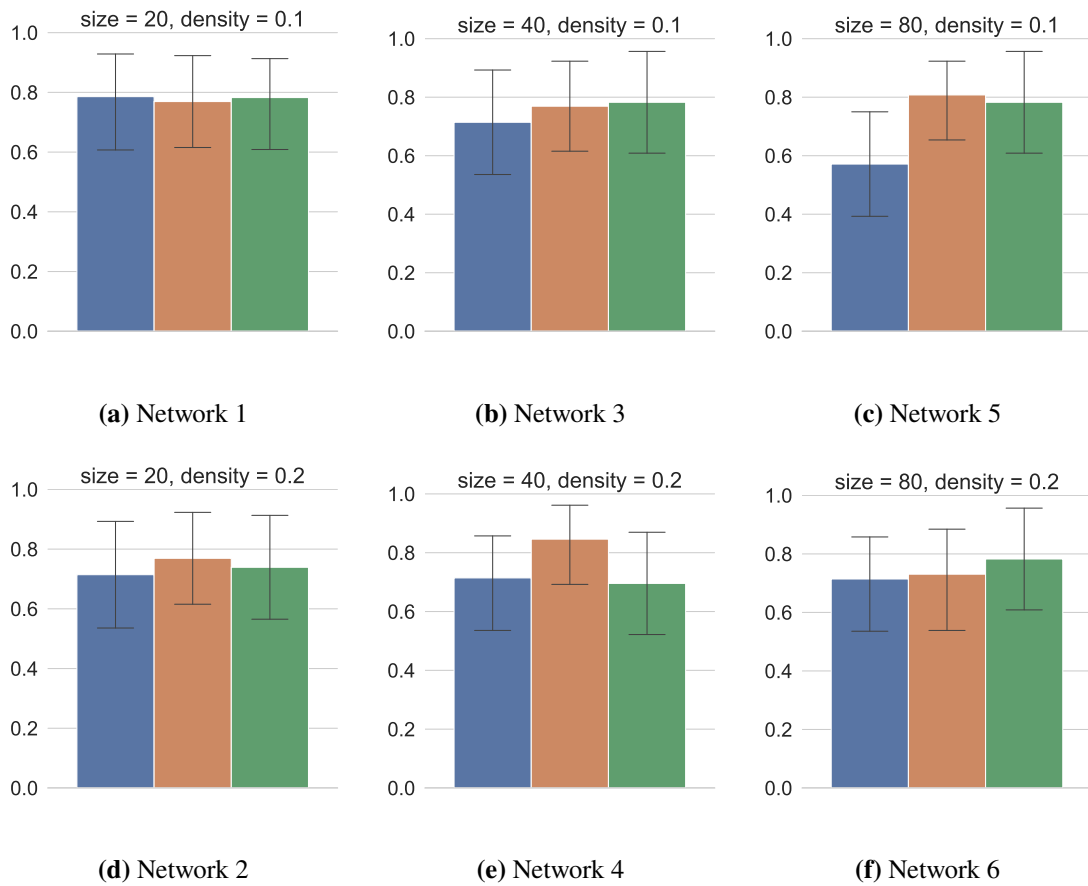


Figure 6.7: Illustration of the accuracy achieved on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for DC task. The network increases in size/density from left to right/top to bottom.

6 Results

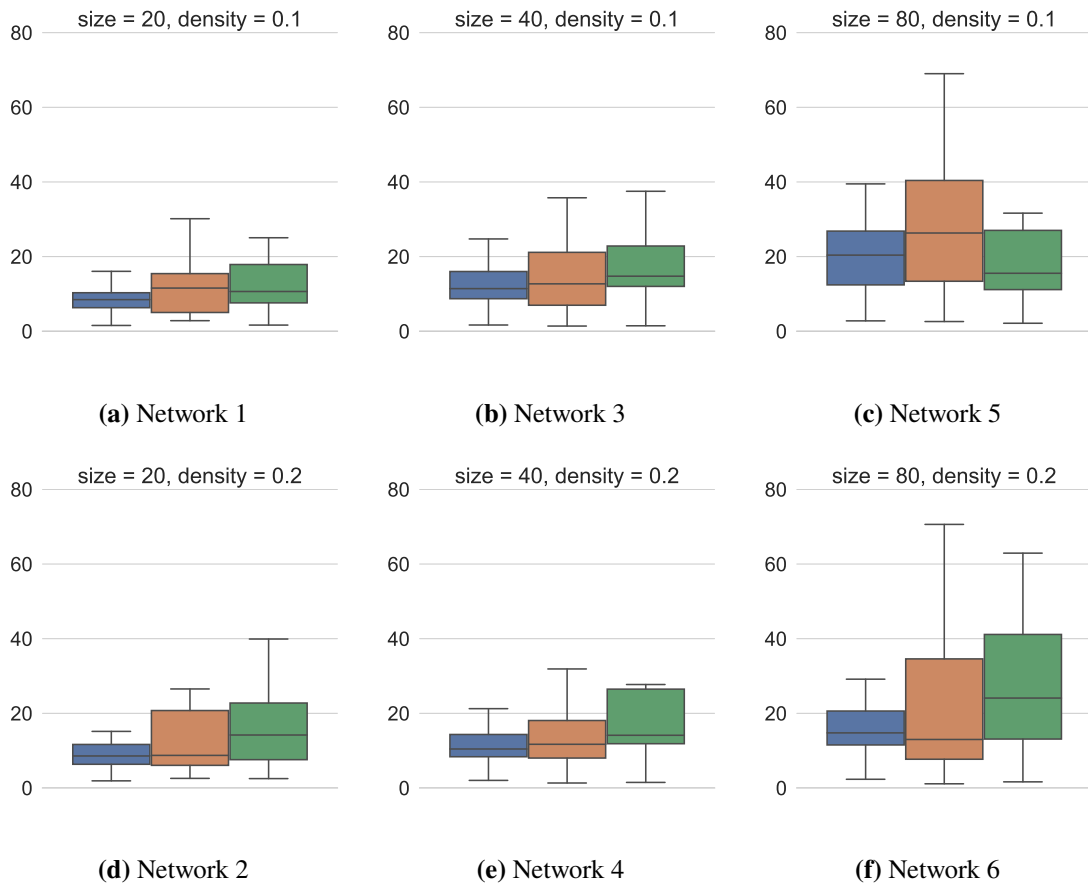


Figure 6.8: Illustration of the distribution of answer time recorded on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for DC task. The network increases in size/density from left to right/top to bottom.

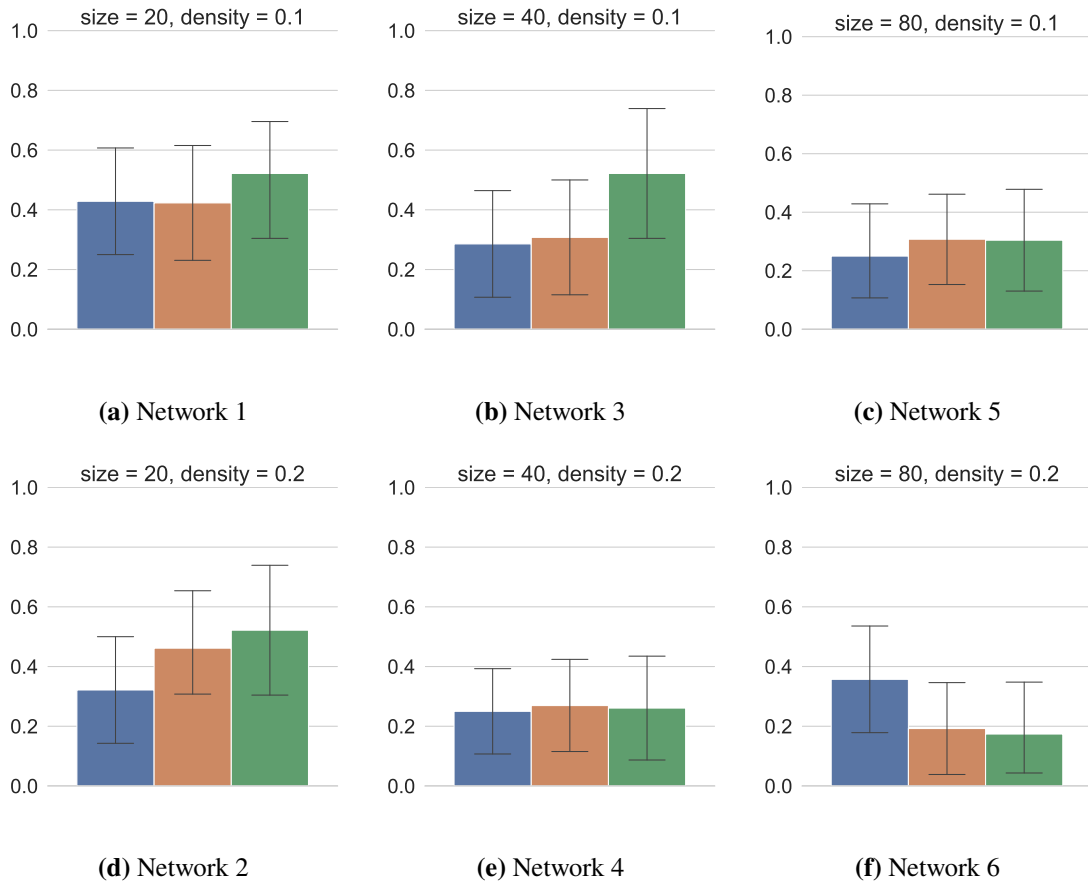


Figure 6.9: Illustration of the accuracy achieved on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for FP task. The network increases in size/density from left to right/top to bottom.

6 Results

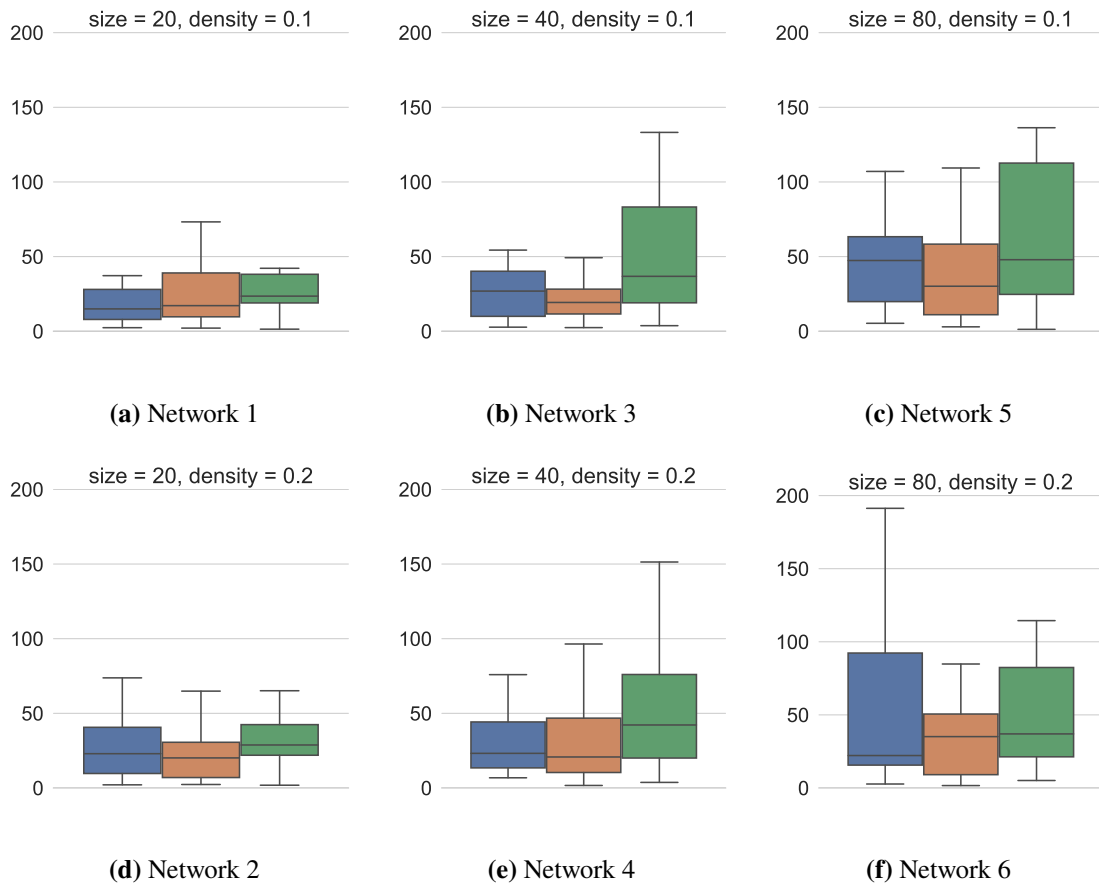


Figure 6.10: Illustration of the distribution of answer time recorded on the respective visualizations - blue: AM, red: VCD, green: ArcM - across each network for FP task. The network increases in size/density from left to right/top to bottom.

7 Discussion

This Chapter is divided into two sections. In Section 7.1, the results from Chapter 6 are put into perspective with respect to the hypotheses in Section 4.1. Section 7.2 lists challenges that were faced during the conduction of the study.

7.1 Discussion of the Results

In the following, for each task, the results are put into perspective for their corresponding hypotheses concerning the accuracy and answer time on the respective visualizations.

7.1.1 Count Neighbors (CN)

The results concerning the CN task (see Section 6.1) are put into perspective for its corresponding hypothesis concerning the accuracy and answer time on the respective visualizations.

Accuracy

Different than hypothesized, the accuracy on the AM is not significantly higher than on any other visualization across all networks. Thus, the rotation of the matrix does not significantly influence the accuracy for this task. It can be argued that scanning in diagonal directions is not as troublesome as originally expected. As hypothesized, no significant difference in accuracy is recorded on the VCD and ArcM, respectively, across all networks. Thus, the visual encoding of links does not significantly influence the accuracy for this task as well.

Answer Time

With one exception, a significantly shorter answer time is recorded on the AM than on any other visualization across all networks, as hypothesized. Thus, the rotation of the matrix leads to an increase in the answer time for this task. It can be argued that scanning in diagonal directions is as time-consuming as originally expected. Different than hypothesized, a significantly shorter answer time is recorded on the VCD than on the ArcM across sparse as well as dense networks. Thus, the visual encoding of links significantly impacts the answer time for this task. Encoding links with arc bends lead to an increase in answer time. It can be argued that arc bends are narrower than squares, and therefore, they need to be counted with great focus.

7.1.2 Identify Link Endpoints (ILE)

The results concerning the ILE task (see Section 6.2) are put into perspective for its corresponding hypothesis concerning the accuracy and answer time on the respective visualizations.

Accuracy

As hypothesized, no significant difference in accuracy is recorded on the respective visualizations across all networks. Thus, the rotation of the matrix does not significantly impact the accuracy for this task.

Answer Time

Different than hypothesized, a significantly shorter answer time is recorded on the AM than on any other visualization across all networks. Thus, the rotation of the matrix leads to an increase in answer time for this task. It can be argued that scanning in diagonal directions is as troublesome for this task as it is for the CN task. No significant difference in answer time is recorded on the VCD and ArcM, respectively, across dense networks. Thus, the visual encoding of links does not significantly impact the answer time for this task. It can be argued that links in a VCD can be as conveniently followed as the links in an ArcM, contradicting the original expectation.

7.1.3 Bidirectional Linkage (BL)

The results concerning the BL task (see Section 6.3) are put into perspective for its corresponding hypothesis concerning the accuracy and answer time on the respective visualizations.

Accuracy

Different than hypothesized, the accuracy achieved on the VCD and ArcM, respectively, is not significantly higher than on the AM across all networks. Thus, the rotation of the matrix does not significantly impact the accuracy for this task. It can be argued that the accuracy in detecting mirror symmetry does not necessarily depend on the alignment of the axis, contradicting the original expectation. As hypothesized, no significant difference in accuracy is recorded on the VCD and ArcM, respectively, across all networks. Thus, the visual encoding of the links does not significantly impact the answer time for this task.

Answer Time

Different than hypothesized, the answer time recorded on the VCD and ArcM, respectively, is not significantly shorter than on the AM across all networks. Thus the rotation of the matrix does not significantly impact the answer time for this task. It can be argued that the time required in detecting mirror symmetry does not necessarily depend on the alignment of the axis, contradicting the original expectation. No significant difference in answer time is recorded on the VCD and

ArcM, respectively, across all networks. Thus, the visual encoding of links does not significantly impact the answer time for this task. It can be argued that “crowded areas” in a visualization do not affect the visual perception of symmetry, indifferent from the visual encoding of links.

7.1.4 Direct Connection (DC)

The results concerning the DC task (see Section 6.4) are put into perspective for its corresponding hypothesis concerning the accuracy and answer time on the respective visualizations.

Accuracy

As hypothesized, no significant difference in accuracy is recorded on any visualization across all networks. Thus, neither the rotation of the matrix nor the visual encoding of links significantly impacts the accuracy for this task.

Answer Time

Different than hypothesized, a significantly shorter answer time is recorded on the AM than on the ArcM across most networks. Thus, the visual encoding of links significantly impacts the answer time. Encoding links with arc bends leads to an increase in answer time. It can be argued that arc bends are narrower than squares, and therefore, they need to be followed with great focus. Moreover, visually extending both sides of an arc bend appears to be difficult to do.

7.1.5 Find Path (FP)

In the following, the results concerning the task (see Section 6.5) are put into perspective for its corresponding hypothesis concerning the accuracy and answer time on the respective visualizations.

Accuracy

Different than hypothesized, the accuracy recorded on the VCD and ArcM, respectively, is not significantly higher than on the AM across all networks. Thus, the rotation of the matrix does not significantly impact the accuracy for this task. It can be argued that the alignment of nodes on the central vertical axis, which results from the rotation of the matrix, does not reduce the deviation of orientation significantly, and thus, users still get confused. Also, the accuracy recorded on the ArcM is not significantly higher than on the VCD across all networks. Thus, the visual encoding of links does not significantly impact the accuracy for this task. It can be argued that visually extending an arc bend is difficult to do, and therefore, it does not reduce the challenge of tracing a connection between two nodes.

Answer Time

Different than hypothesized, the answer time recorded on the VCD and ArcM, respectively, is not significantly higher than on the AM across all networks. Thus, the rotation of the matrix does not significantly impact the answer time for this task. It can be argued that the alignment of nodes on the central vertical axis, which results from the rotation of the matrix, does not reduce the deviation of orientation significantly. Thus, users rely on the “trial and error” strategy, which is very time-consuming. Also, the answer time recorded on the ArcM is not significantly higher than on the VCD across all networks. Thus, the visual encoding of links does not significantly impact the answer time for this task. It can be argued that visually extending an arc bend is time-consuming, so that users revert back to the “trial and error” strategy.

7.2 Challenges

The study was conducted at the time where the COVID-19 pandemic already reached Germany ¹. During this period, preventive measures were introduced to reduce the chances of infection which include staying at home, avoiding crowded places, and keeping distance from others. The imposed restrictions naturally affected the study in the way that it could not be conducted under highly controlled conditions (e.g., laboratory). Therefore, the study was outsourced to MTurk to make it accessible to people all around the world. A great advantage of a crowd-sourced study over a controlled lab study is that more participants can be gathered in a relatively short time. Also, chances are high that these participants are diverse not only in regards to age, gender, and education but also in regards to the level of experience with network visualization, level of knowledge regarding network analysis, and familiarity with the term “Adjacency Matrix”. However, there are also three drawbacks of a crowd-sourced study that became apparent during the study. The first drawback is that not all participants take the study seriously, meaning that they are only motivated by the money they receive at the end of the study. Several such participants were identified when reviewing the submissions of each participant. It was possible to categorize these participants into two groups. One group of participants speedran through the study, meaning they did not spend much time on either reading the introductory parts of the study or the tasks. The other group of participants did not perform the tasks seriously. Whether participants performed the tasks seriously or not was identified by scanning over their answers provided in the tasks, particularly in the CN and ILE tasks. In the CN task, either the same single-digit or node X were provided as the answer throughout all the trials, while in the ILE task, the answer in each trial consisted either of the same pair of nodes X and Y or any random combination of node pairs. However, even after filtering out participants that did not take the study seriously based on the two aforementioned “filtering criteria”, still, several submissions contained inexplicable and inconsistent answers in regards to rating the perceived difficulty of each task. Feedback in regards to how they approached the task rarely contained qualitative answers. Mostly, feedback consisted of uninformative one-liners, such as “I followed the instructions”. The second drawback concerns MTurk’s inability to identify the wrong codes. A code is required to confirm the participation of a user in the study. Wrong codes were encountered in form of random numbers or participant’s worker ID. If MTurk was able to identify wrong codes, then less time has to be spent on double-checking each code. Both drawbacks lead to the fact that

¹https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Steckbrief.html

more time is spent on running the study than required. The last drawback concerns the scenario where participants have a question in regards to anything related to the study, but the answer can not be found in the pages of the study. Therefore, participants have to guess the answer or they quit. One solution to this problem is that participants contact the creator of the study through email, but this is a lengthy process. Nevertheless, the study was conducted successfully. For crowd-sourced studies that are going to be conducted in the future, it is recommended to contact each participant through a video communications platform before they take the study. The participants can be guided through the procedure of the study thoroughly, and thus, the importance of the study itself is made clear to each participant, so that each of them takes the study seriously. If questions arise during the study, they can be answered on the spot.

8 Conclusion

In this work, reasons for the AM's unsuitability for path-related tasks were identified. The first problem concerns the laborious orientation in an AM when performing path-related tasks on the said visualization. The second problem concerns the non-intuitive encoding of links in an AM. Consequently, solutions to the problems were proposed with the introduction of several variants of AM. Two variants of AM, namely, *Vertical Cross-Diagram* (VCD) and *Horizontal Cross-Diagram* (HCD), were proposed with the focus on the rotation of the traditional matrix, to solve the first problem mentioned above. Another variant of AM, namely, *Single Curved Polyline Adjacency Matrix* (SCPAM) was proposed with the focus on the visual encoding of links, to solve the second problem mentioned above. The last two variants, namely, *Arc Diagram* (ArcD) and *Arc Matrix* (ArcM), were proposed with the focus on the rotation of the matrix as well as the visual encoding of links, to solve both problems mentioned above. The goal of this thesis was to compare two of the proposed variants, namely, VCD and ArcM, with the AM for several network analysis tasks, to investigate how the rotation of the matrix and the visual encoding of links influence the user performance at those tasks, particularly path-related tasks. To this end, a crowd-sourced study was conducted, where participants were exposed to one of the visualizations (i.e., AM, VCD, or ArcM) on that they performed a variety of network analysis tasks across networks differing in size and density. It was hypothesized that the rotation of the matrix, as well as the visual encoding of links, would significantly increase user performance not only at several connectivity-related tasks but also at the path-finding task. The results reveal that the accuracy is not significantly affected by the rotation of the matrix, as well as the visual encoding of links, across all tasks. This means that users achieve similar accuracy on all visualizations, indifferent of the rotation of the matrix or the visual encoding of links. Several isolated cases exist, where both, the VCD and ArcM record a visibly higher accuracy than the AM. However, the difference in accuracy is deemed not significant in all those cases. For example, visibly higher accuracy is recorded on the VCD as well as on the ArcM than on the AM for the task concerning the detection of mirror symmetry, across most networks. However, the difference in accuracy is deemed not significant across those networks. The results also reveal that the answer time is a critical factor. The rotation of the matrix, as well as the visual encoding of links, lead to significant degradation of answer time across most connectivity-related tasks. As for the path-finding task, the answer time is not significantly affected by both, the rotation of the matrix and the visual encoding of links.

Future Work

This work can be further extended by evaluating the variants that were not covered in the study, namely, HCD, SCPAM, and ArcD. The evaluation of the SCPAM and ArcD, respectively, is interesting insofar that in both variants, links are encoded with lines that connect their endpoints directly. It is interesting to investigate, how the visual encoding of links not only affects the user

performance at path-related tasks but also how it affects the user performance if the underlying network increases in density. It is also interesting to investigate, how novel hybrid approaches such as MatLink [HF07] or NodeTrix [HFM07] compare with the proposed variants in this work, under the condition that the underlying network is directed.

A Calculation of P-values for Network Property Influence Testing

A.1 Count Neighbors (CN)

Vis.	Network 1 (mean)	Network 2 (mean)	Network 3 (mean)	Network 4 (mean)	Network 5 (mean)	Network 6 (mean)
AM	79%	71%	64%	64%	64%	50%
VCD	46%	54%	54%	50%	38%	27%
ArcM	65%	65%	61%	48%	61%	26%

Table A.1: Accuracy achieved on each visualization across each network for CN task.

Network A	Network B	p-value
Network 1	Network 3	0.244
Network 2	Network 4	0.575
Network 3	Network 5	1.000
Network 4	Network 6	0.288

(a) AM

Network A	Network B	p-value
Network 1	Network 3	0.588
Network 2	Network 4	0.786
Network 3	Network 5	0.275
Network 4	Network 6	0.090

(b) VCD

Network A	Network B	p-value
Network 1	Network 3	0.766
Network 2	Network 4	0.244
Network 3	Network 5	1.000
Network 4	Network 6	0.132

(c) ArcM

Table A.2: Difference in accuracy achieved on one visualization across two networks, respectively, varying in size for CN task. Significant difference (according to Welch's t-test) highlighted in bold.

A Calculation of P-values for Network Property Influence Testing

Network A	Network B	p-value
Network 1	Network 2	0.546
Network 3	Network 4	1.000
Network 5	Network 6	0.288

(a) AM

Network A	Network B	p-value
Network 1	Network 2	0.588
Network 3	Network 4	0.786
Network 5	Network 6	0.385

(b) VCD

Network A	Network B	p-value
Network 1	Network 2	1.000
Network 3	Network 4	0.386
Network 5	Network 6	0.017

(c) ArcM

Table A.3: Difference in accuracy achieved on one visualization across two networks, respectively, varying in density for CN task. Significant difference (according to Welch’s t-test) highlighted in bold.

Vis.	Network 1 (mean)	Network 2 (mean)	Network 3 (mean)	Network 4 (mean)	Network 5 (mean)	Network 6 (mean)
AM	10.46s	12.85s	17.32s	23.49s	29.42s	36.48s
VCD	11.67s	11.73s	17.46s	20.46s	33.90s	44.95s
ArcM	14.95s	18.99s	21.95s	28.76s	47.43s	57.07s

Table A.4: Mean answer time recorded on each visualization across each network for CN task.

Network A	Network B	p-value
Network 1	Network 3	≈ 0.000
Network 2	Network 4	≈ 0.000
Network 3	Network 5	≈ 0.000
Network 4	Network 6	0.002

(a) AM

Network A	Network B	p-value
Network 1	Network 3	0.008
Network 2	Network 4	0.002
Network 3	Network 5	0.002
Network 4	Network 6	0.002

(b) VCD

Network A	Network B	p-value
Network 1	Network 3	0.001
Network 2	Network 4	0.013
Network 3	Network 5	≈ 0.000
Network 4	Network 6	0.001

(c) ArcM

Table A.5: Difference in answer time recorded on one visualization across two networks, respectively, varying in size for CN task. Significant difference (according to Welch’s t-test) highlighted in bold.

Network A	Network B	p-value
Network 1	Network 2	0.106
Network 3	Network 4	0.029
Network 5	Network 6	0.100

(a) AM

Network A	Network B	p-value
Network 1	Network 2	0.973
Network 3	Network 4	0.316
Network 5	Network 6	0.180

(b) VCD

Network A	Network B	p-value
Network 1	Network 2	0.112
Network 3	Network 4	0.057
Network 5	Network 6	0.296

(c) ArcM

Table A.6: Difference in answer time recorded on one visualization across two networks, respectively, varying in density for CN task. Significant difference (according to Welch’s t-test) highlighted in bold.

A.2 Identify Link Endpoints (ILE)

Vis.	Network 1 (mean)	Network 2 (mean)	Network 3 (mean)	Network 4 (mean)	Network 5 (mean)	Network 6 (mean)
AM	100%	96%	100%	96%	89%	89%
VCD	85%	81%	92%	88%	81%	88%
ArcM	87%	91%	83%	96%	87%	91%

Table A.7: Accuracy achieved on each visualization across each network for ILE task.

A Calculation of P-values for Network Property Influence Testing

Network A	Network B	p-value
Network 1	Network 3	1.000
Network 2	Network 4	1.000
Network 3	Network 5	0.083
Network 4	Network 6	0.309

(a) AM

Network A	Network B	p-value
Network 1	Network 3	0.396
Network 2	Network 4	0.452
Network 3	Network 5	0.232
Network 4	Network 6	1.000

(b) VCD

Network A	Network B	p-value
Network 1	Network 3	0.689
Network 2	Network 4	0.561
Network 3	Network 5	0.689
Network 4	Network 6	0.561

(c) ArcM

Table A.8: Difference in accuracy achieved on one visualization across two networks, respectively, varying in size for ILE task. Significant difference (according to Welch's t-test) highlighted in bold.

Network A	Network B	p-value
Network 1	Network 2	0.326
Network 3	Network 4	0.326
Network 5	Network 6	1.000

(a) AM

Network A	Network B	p-value
Network 1	Network 2	0.720
Network 3	Network 4	0.646
Network 5	Network 6	0.452

(b) VCD

Network A	Network B	p-value
Network 1	Network 2	0.645
Network 3	Network 4	0.164
Network 5	Network 6	0.645

(c) ArcM

Table A.9: Difference in accuracy achieved on one visualization across two networks, respectively, varying in density for ILE task. Significant difference (according to Welch's t-test) highlighted in bold.

Vis.	Network 1 (mean)	Network 2 (mean)	Network 3 (mean)	Network 4 (mean)	Network 5 (mean)	Network 6 (mean)
AM	7.62s	7.93s	10.44s	10.18s	17.58s	15.67s
VCD	11.83s	12.66s	18.60s	15.65s	26.22s	26.89s
ArcM	15.03s	11.37s	19.02s	19.62s	29.29s	31.71s

Table A.10: Mean answer time recorded on each visualization across each network for ILE task.

Network A	Network B	p-value
Network 1	Network 3	\approx 0.000
Network 2	Network 4	0.001
Network 3	Network 5	\approx 0.000
Network 4	Network 6	\approx 0.000

(a) AM

Network A	Network B	p-value
Network 1	Network 3	0.001
Network 2	Network 4	0.113
Network 3	Network 5	0.035
Network 4	Network 6	0.005

(b) VCD

Network A	Network B	p-value
Network 1	Network 3	0.155
Network 2	Network 4	0.008
Network 3	Network 5	0.011
Network 4	Network 6	0.008

(c) ArcM

Table A.11: Difference in answer time recorded on one visualization across two networks, respectively, varying in size for ILE task. Significant difference (according to Welch's t-test) highlighted in bold.

Network A	Network B	p-value
Network 1	Network 2	0.565
Network 3	Network 4	0.749
Network 5	Network 6	0.256

(a) AM

Network A	Network B	p-value
Network 1	Network 2	0.503
Network 3	Network 4	0.219
Network 5	Network 6	0.882

(b) VCD

Network A	Network B	p-value
Network 1	Network 2	0.063
Network 3	Network 4	0.863
Network 5	Network 6	0.603

(c) ArcM

Table A.12: Difference in answer time recorded on one visualization across two networks, respectively, varying in density for ILE task. Significant difference (according to Welch's t-test) highlighted in bold.

A.3 Bidirectional Linkage (BL)

Vis.	Network 1 (mean)	Network 2 (mean)	Network 3 (mean)	Network 4 (mean)	Network 5 (mean)	Network 6 (mean)
AM	68%	61%	54%	64%	50%	36%
VCD	62%	65%	62%	69%	38%	73%
ArcM	74%	74%	83%	91%	52%	48%

Table A.13: Accuracy achieved on each visualization across each network for BL task.

Network A	Network B	p-value
Network 1	Network 3	0.282
Network 2	Network 4	0.787
Network 3	Network 5	0.794
Network 4	Network 6	0.033

(a) AM

Network A	Network B	p-value
Network 1	Network 3	1.000
Network 2	Network 4	0.773
Network 3	Network 5	0.099
Network 4	Network 6	0.765

(b) VCD

Network A	Network B	p-value
Network 1	Network 3	0.486
Network 2	Network 4	0.126
Network 3	Network 5	0.028
Network 4	Network 6	0.001

(c) ArcM

Table A.14: Difference in accuracy achieved on one visualization across two networks, respectively, varying in size for BL task. Significant difference (according to Welch's t-test) highlighted in bold.

Network A	Network B	p-value
Network 1	Network 2	0.585
Network 3	Network 4	0.424
Network 5	Network 6	0.288

(a) AM

Network A	Network B	p-value
Network 1	Network 2	0.779
Network 3	Network 4	0.569
Network 5	Network 6	0.011

(b) VCD

Network A	Network B	p-value
Network 1	Network 2	1.000
Network 3	Network 4	0.393
Network 5	Network 6	0.774

(c) ArcM

Table A.15: Difference in accuracy achieved on one visualization across two networks, respectively, varying in density for BL task. Significant difference (according to Welch's t-test) highlighted in bold.

Vis.	Network 1 (mean)	Network 2 (mean)	Network 3 (mean)	Network 4 (mean)	Network 5 (mean)	Network 6 (mean)
AM	2.52s	2.59s	2.53s	2.44s	2.64s	3.30s
VCD	2.31s	2.30s	2.38s	2.55s	2.87s	2.66s
ArcM	2.15s	2.26s	2.60s	2.25s	2.79s	2.47s

Table A.16: Mean answer time recorded on each visualization across each network for BL task.

Network A	Network B	p-value
Network 1	Network 3	0.971
Network 2	Network 4	0.547
Network 3	Network 5	0.590
Network 4	Network 6	0.007

(a) AM

Network A	Network B	p-value
Network 1	Network 3	0.686
Network 2	Network 4	0.447
Network 3	Network 5	0.097
Network 4	Network 6	0.760

(b) VCD

Network A	Network B	p-value
Network 1	Network 3	0.101
Network 2	Network 4	0.934
Network 3	Network 5	0.575
Network 4	Network 6	0.290

(c) ArcM

Table A.17: Difference in answer time recorded on one visualization across two networks, respectively, varying in size for BL task. Significant difference (according to Welch's t-test) highlighted in bold.

Network A	Network B	p-value
Network 1	Network 2	0.780
Network 3	Network 4	0.702
Network 5	Network 6	0.032

(a) AM

Network A	Network B	p-value
Network 1	Network 2	0.984
Network 3	Network 4	0.579
Network 5	Network 6	0.505

(b) VCD

Network A	Network B	p-value
Network 1	Network 2	0.561
Network 3	Network 4	0.158
Network 5	Network 6	0.325

(c) ArcM

Table A.18: Difference in answer time recorded on one visualization across two networks, respectively, varying in density for BL task. Significant difference (according to Welch's t-test) highlighted in bold.

A.4 Direct Connection (DC)

Vis.	Network 1 (mean)	Network 2 (mean)	Network 3 (mean)	Network 4 (mean)	Network 5 (mean)	Network 6 (mean)
AM	79%	71%	71%	71%	57%	71%
VCD	77%	77%	77%	85%	81%	73%
ArcM	78%	74%	78%	70%	78%	78%

Table A.19: Accuracy achieved on each visualization across each network for DC task.

Network A	Network B	p-value
Network 1	Network 3	0.546
Network 2	Network 4	1.000
Network 3	Network 5	0.273
Network 4	Network 6	1.000

(a) AM

Network A	Network B	p-value
Network 1	Network 3	1.000
Network 2	Network 4	0.491
Network 3	Network 5	0.740
Network 4	Network 6	0.318

(b) VCD

Network A	Network B	p-value
Network 1	Network 3	1.000
Network 2	Network 4	0.750
Network 3	Network 5	1.000
Network 4	Network 6	0.513

(c) ArcM

Table A.20: Difference in accuracy achieved on one visualization across two networks, respectively, varying in size for DC task. Significant difference (according to Welch's t-test) highlighted in bold.

Network A	Network B	p-value
Network 1	Network 2	0.546
Network 3	Network 4	1.000
Network 5	Network 6	0.273

(a) AM

Network A	Network B	p-value
Network 1	Network 2	1.000
Network 3	Network 4	0.491
Network 5	Network 6	0.519

(b) VCD

Network A	Network B	p-value
Network 1	Network 2	0.737
Network 3	Network 4	0.513
Network 5	Network 6	1.000

(c) ArcM

Table A.21: Difference in accuracy achieved on one visualization across two networks, respectively, varying in density for DC task. Significant difference (according to Welch's t-test) highlighted in bold.

Vis.	Network 1 (mean)	Network 2 (mean)	Network 3 (mean)	Network 4 (mean)	Network 5 (mean)	Network 6 (mean)
AM	8.33s	8.30s	11.72s	10.93s	18.54s	14.60s
VCD	11.58s	9.73s	12.39s	12.12s	26.07s	19.88s
ArcM	11.04s	15.47s	16.64s	15.82s	15.06s	24.90s

Table A.22: Mean answer time recorded on each visualization across each network for DC task.

Network A	Network B	p-value
Network 1	Network 3	0.009
Network 2	Network 4	0.025
Network 3	Network 5	0.005
Network 4	Network 6	0.029

(a) AM

Network A	Network B	p-value
Network 1	Network 3	0.741
Network 2	Network 4	0.283
Network 3	Network 5	0.004
Network 4	Network 6	0.073

(b) VCD

Network A	Network B	p-value
Network 1	Network 3	0.024
Network 2	Network 4	0.907
Network 3	Network 5	0.567
Network 4	Network 6	0.052

(c) ArcM

Table A.23: Difference in answer time recorded on one visualization across two networks, respectively, varying in size for DC task. Significant difference (according to Welch's t-test) highlighted in bold.

Network A	Network B	p-value
Network 1	Network 2	0.978
Network 3	Network 4	0.576
Network 5	Network 6	0.111

(a) AM

Network A	Network B	p-value
Network 1	Network 2	0.361
Network 3	Network 4	0.917
Network 5	Network 6	0.271

(b) VCD

Network A	Network B	p-value
Network 1	Network 2	0.109
Network 3	Network 4	0.766
Network 5	Network 6	0.036

(c) ArcM

Table A.24: Difference in answer time recorded on one visualization across two networks, respectively, varying in density for DC task. Significant difference (according to Welch's t-test) highlighted in bold.

A.5 Find Path (FP)

Vis.	Network 1 (mean)	Network 2 (mean)	Network 3 (mean)	Network 4 (mean)	Network 5 (mean)	Network 6 (mean)
AM	43%	32%	29%	25%	25%	36%
VCD	42%	46%	31%	27%	31%	19%
ArcM	52%	52%	52%	26%	30%	17%

Table A.25: Accuracy achieved on each visualization across each network for FP task.

Network A	Network B	p-value
Network 1	Network 3	0.273
Network 2	Network 4	0.562
Network 3	Network 5	0.768
Network 4	Network 6	0.392

(a) AM

Network A	Network B	p-value
Network 1	Network 3	0.397
Network 2	Network 4	0.156
Network 3	Network 5	1.000
Network 4	Network 6	0.519

(b) VCD

Network A	Network B	p-value
Network 1	Network 3	1.000
Network 2	Network 4	0.073
Network 3	Network 5	0.140
Network 4	Network 6	0.486

(c) ArcM

Table A.26: Difference in accuracy achieved on one visualization across two networks, respectively, varying in size for FP task. Significant difference (according to Welch's t-test) highlighted in bold.

Network A	Network B	p-value
Network 1	Network 2	0.417
Network 3	Network 4	0.768
Network 5	Network 6	0.392

(a) AM

Network A	Network B	p-value
Network 1	Network 2	0.785
Network 3	Network 4	0.765
Network 5	Network 6	0.346

(b) VCD

Network A	Network B	p-value
Network 1	Network 2	1.000
Network 3	Network 4	0.072
Network 5	Network 6	0.310

(c) ArcM

Table A.27: Difference in accuracy achieved on one visualization across two networks, respectively, varying in density for FP task. Significant difference (according to Welch's t-test) highlighted in bold.

Vis.	Network 1 (mean)	Network 2 (mean)	Network 3 (mean)	Network 4 (mean)	Network 5 (mean)	Network 6 (mean)
AM	15.10s	27.94s	23.76s	26.50s	39.46s	50.05s
VCD	25.39s	20.09s	20.72s	29.65s	36.32s	33.61s
ArcM	22.63s	30.71s	50.24s	49.26s	53.91s	43.61s

Table A.28: Mean answer time recorded on each visualization across each network for FP task.

Network A	Network B	p-value
Network 1	Network 3	0.025
Network 2	Network 4	0.806
Network 3	Network 5	0.025
Network 4	Network 6	0.043

(a) AM

Network A	Network B	p-value
Network 1	Network 3	0.364
Network 2	Network 4	0.134
Network 3	Network 5	0.028
Network 4	Network 6	0.583

(b) VCD

Network A	Network B	p-value
Network 1	Network 3	0.004
Network 2	Network 4	0.029
Network 3	Network 5	0.775
Network 4	Network 6	0.589

(c) ArcM

Table A.29: Difference in answer time recorded on one visualization across two networks, respectively, varying in size for FP task. Significant difference (according to Welch's t-test) highlighted in bold.

Network A	Network B	p-value
Network 1	Network 2	0.013
Network 3	Network 4	0.577
Network 5	Network 6	0.385

(a) AM

Network A	Network B	p-value
Network 1	Network 2	0.341
Network 3	Network 4	0.137
Network 5	Network 6	0.734

(b) VCD

Network A	Network B	p-value
Network 1	Network 2	0.069
Network 3	Network 4	0.930
Network 5	Network 6	0.398

(c) ArcM

Table A.30: Difference in answer time recorded on one visualization across two networks, respectively, varying in density for FP task. Significant difference (according to Welch's t-test) highlighted in bold.

B Calculation of P-values for Hypothesis Significance Testing

B.1 Count Neighbors (CN)

Vis. A (mean)	Vis. B (mean)	p-value
AM (79%)	VCD (46%)	0.014
AM (79%)	ArcM (65%)	0.305
VCD (46%)	ArcM (65%)	0.187

(a) Network 1

Vis. A (mean)	Vis. B (mean)	p-value
AM (71%)	VCD (54%)	0.189
AM (71%)	ArcM (65%)	0.644
VCD (54%)	ArcM (65%)	0.428

(b) Network 2

Vis. A (mean)	Vis. B (mean)	p-value
AM (64%)	VCD (54%)	0.446
AM (64%)	ArcM (61%)	0.807
VCD (54%)	ArcM (61%)	0.628

(c) Network 3

Vis. A (mean)	Vis. B (mean)	p-value
AM (64%)	VCD (50%)	0.298
AM (64%)	ArcM (48%)	0.249
VCD (50%)	ArcM (48%)	0.882

(d) Network 4

Vis. A (mean)	Vis. B (mean)	p-value
AM (64%)	VCD (38%)	0.059
AM (64%)	ArcM (61%)	0.806
VCD (38%)	ArcM (61%)	0.122

(e) Network 5

Vis. A (mean)	Vis. B (mean)	p-value
AM (50%)	VCD (27%)	0.084
AM (50%)	ArcM (26%)	0.081
VCD (27%)	ArcM (26%)	0.948

(f) Network 6

Table B.1: Difference in accuracy achieved on two different visualizations, respectively, across each network for CN task. Significant difference (according to Welch's t-test) highlighted in bold.

Vis. A (mean)	Vis. B (mean)	p-value
AM (10.46s)	VCD (11.67s)	0.424
AM (10.46s)	ArcM (14.95s)	0.019
VCD (11.67s)	ArcM (14.95s)	0.112

(a) Network 1

Vis. A (mean)	Vis. B (mean)	p-value
AM (12.85s)	VCD (11.73s)	0.493
AM (12.85s)	ArcM (18.99s)	0.009
VCD (11.73s)	ArcM (18.99s)	0.002

(b) Network 2

Vis. A (mean)	Vis. B (mean)	p-value
AM (17.32s)	VCD (17.46s)	0.948
AM (17.32s)	ArcM (21.95s)	0.018
VCD (17.46s)	ArcM (21.95s)	0.034

(c) Network 3

Vis. A (mean)	Vis. B (mean)	p-value
AM (23.49s)	VCD (20.46s)	0.368
AM (23.49s)	ArcM (28.76s)	0.191
VCD (20.46s)	ArcM (28.76s)	0.046

(d) Network 4

Vis. A (mean)	Vis. B (mean)	p-value
AM (29.42s)	VCD (33.90s)	0.398
AM (29.42s)	ArcM (47.43s)	0.005
VCD (33.90s)	ArcM (47.43s)	0.057

(e) Network 5

Vis. A (mean)	Vis. B (mean)	p-value
AM (36.48s)	VCD (44.95s)	0.266
AM (36.48s)	ArcM (57.07s)	0.015
VCD (44.95s)	ArcM (57.07s)	0.233

(f) Network 6

Table B.2: Difference in answer time recorded on two different visualizations, respectively, across each network for CN task. Significant difference (according to Welch's t-test) highlighted in bold.

B.2 Identify Link Endpoints (ILE)

Vis. A (mean)	Vis. B (mean)	p-value
AM (100%)	VCD (85%)	0.043
AM (100%)	ArcM (87%)	0.083
VCD (85%)	ArcM (87%)	0.819

(a) Network 1

Vis. A (mean)	Vis. B (mean)	p-value
AM (96%)	VCD (81%)	0.079
AM (96%)	ArcM (91%)	0.468
VCD (81%)	ArcM (91%)	0.293

(b) Network 2

Vis. A (mean)	Vis. B (mean)	p-value
AM (100%)	VCD (92%)	0.161
AM (100%)	ArcM (83%)	0.043
VCD (92%)	ArcM (83%)	0.322

(c) Network 3

Vis. A (mean)	Vis. B (mean)	p-value
AM (96%)	VCD (88%)	0.283
AM (96%)	ArcM (96%)	0.891
VCD (88%)	ArcM (96%)	0.357

(d) Network 4

Vis. A (mean)	Vis. B (mean)	p-value
AM (89%)	VCD (81%)	0.393
AM (89%)	ArcM (87%)	0.804
VCD (81%)	ArcM (87%)	0.564

(e) Network 5

Vis. A (mean)	Vis. B (mean)	p-value
AM (89%)	VCD (88%)	0.925
AM (89%)	ArcM (91%)	0.812
VCD (88%)	ArcM (91%)	0.747

(f) Network 6

Table B.3: Difference in accuracy achieved on two different visualizations, respectively, across each network for ILE task. Significant difference (according to Welch's t-test) highlighted in bold.

Vis. A (mean)	Vis. B (mean)	p-value
AM (7.62s)	VCD (11.83s)	\approx 0.000
AM (7.62s)	ArcM (15.03s)	\approx 0.000
VCD (11.83s)	ArcM (15.03s)	0.093

(a) Network 1

Vis. A (mean)	Vis. B (mean)	p-value
AM (7.93s)	VCD (12.66s)	\approx 0.000
AM (7.93s)	ArcM (11.37s)	0.001
VCD (12.66s)	ArcM (11.37s)	0.331

(b) Network 2

Vis. A (mean)	Vis. B (mean)	p-value
AM (10.44s)	VCD (18.60s)	\approx 0.000
AM (10.44s)	ArcM (19.02s)	0.001
VCD (18.60s)	ArcM (19.02s)	0.884

(c) Network 3

Vis. A (mean)	Vis. B (mean)	p-value
AM (10.18s)	VCD (15.65s)	0.002
AM (10.18s)	ArcM (19.62s)	0.003
VCD (15.65s)	ArcM (19.62s)	0.215

(d) Network 4

Vis. A (mean)	Vis. B (mean)	p-value
AM (17.58s)	VCD (26.22s)	0.012
AM (17.58s)	ArcM (29.29s)	0.002
VCD (26.22s)	ArcM (29.29s)	0.487

(e) Network 5

Vis. A (mean)	Vis. B (mean)	p-value
AM (15.67s)	VCD (26.89s)	0.004
AM (15.67s)	ArcM (31.71s)	\approx 0.000
VCD (26.89s)	ArcM (31.71s)	0.317

(f) Network 6

Table B.4: Difference in answer time recorded on two different visualizations, respectively, across each network for ILE task. Significant difference (according to Welch's t-test) highlighted in bold.

B.3 Bidirectional Linkage (BL)

Vis. A (mean)	Vis. B (mean)	p-value
AM (68%)	VCD (62%)	0.635
AM (68%)	ArcM (74%)	0.643
VCD (62%)	ArcM (74%)	0.364

(a) Network 1

Vis. A (mean)	Vis. B (mean)	p-value
AM (61%)	VCD (65%)	0.728
AM (61%)	ArcM (74%)	0.325
VCD (65%)	ArcM (74%)	0.526

(b) Network 2

Vis. A (mean)	Vis. B (mean)	p-value
AM (54%)	VCD (62%)	0.562
AM (54%)	ArcM (83%)	0.025
VCD (62%)	ArcM (83%)	0.102

(c) Network 3

Vis. A (mean)	Vis. B (mean)	p-value
AM (64%)	VCD (69%)	0.706
AM (64%)	ArcM (91%)	0.018
VCD (69%)	ArcM (91%)	0.051

(d) Network 4

Vis. A (mean)	Vis. B (mean)	p-value
AM (50%)	VCD (38%)	0.403
AM (50%)	ArcM (52%)	0.880
VCD (38%)	ArcM (52%)	0.347

(e) Network 5

Vis. A (mean)	Vis. B (mean)	p-value
AM (36%)	VCD (73%)	0.005
AM (36%)	ArcM (48%)	0.394
VCD (73%)	ArcM (48%)	0.075

(f) Network 6

Table B.5: Difference in accuracy achieved on two different visualizations, respectively, across each network for BL task. Significant difference (according to Welch's t-test) highlighted in bold.

Vis. A (mean)	Vis. B (mean)	p-value
AM (2.52s)	VCD (2.31s)	0.293
AM (2.52s)	ArcM (2.15s)	0.086
VCD (2.31s)	ArcM (2.15s)	0.472

(a) Network 1

Vis. A (mean)	Vis. B (mean)	p-value
AM (2.59s)	VCD (2.30s)	0.291
AM (2.59s)	ArcM (2.26s)	0.163
VCD (2.30s)	ArcM (2.26s)	0.835

(b) Network 2

Vis. A (mean)	Vis. B (mean)	p-value
AM (2.53s)	VCD (2.38s)	0.512
AM (2.53s)	ArcM (2.60s)	0.812
VCD (2.38s)	ArcM (2.60s)	0.389

(c) Network 3

Vis. A (mean)	Vis. B (mean)	p-value
AM (2.44s)	VCD (2.55s)	0.720
AM (2.44s)	ArcM (2.25s)	0.256
VCD (2.55s)	ArcM (2.25s)	0.318

(d) Network 4

Vis. A (mean)	Vis. B (mean)	p-value
AM (2.64s)	VCD (2.87s)	0.425
AM (2.64s)	ArcM (2.79s)	0.619
VCD (2.87s)	ArcM (2.79s)	0.822

(e) Network 5

Vis. A (mean)	Vis. B (mean)	p-value
AM (3.30s)	VCD (2.66s)	0.058
AM (3.30s)	ArcM (2.47s)	0.014
VCD (2.66s)	ArcM (2.47s)	0.475

(f) Network 6

Table B.6: Difference in answer time recorded on two different visualizations, respectively, across each network for BL task. Significant difference (according to Welch's t-test) highlighted in bold.

B.4 Direct Connection (DC)

Vis. A (mean)	Vis. B (mean)	p-value
AM (79%)	VCD (77%)	0.887
AM (79%)	ArcM (78%)	0.979
VCD (77%)	ArcM (78%)	0.913

(a) Network 1

Vis. A (mean)	Vis. B (mean)	p-value
AM (71%)	VCD (77%)	0.652
AM (71%)	ArcM (74%)	0.847
VCD (77%)	ArcM (74%)	0.812

(b) Network 2

Vis. A (mean)	Vis. B (mean)	p-value
AM (71%)	VCD (77%)	0.652
AM (71%)	ArcM (78%)	0.583
VCD (77%)	ArcM (78%)	0.913

(c) Network 3

Vis. A (mean)	Vis. B (mean)	p-value
AM (71%)	VCD (85%)	0.248
AM (71%)	ArcM (70%)	0.887
VCD (85%)	ArcM (70%)	0.223

(d) Network 4

Vis. A (mean)	Vis. B (mean)	p-value
AM (57%)	VCD (81%)	0.062
AM (57%)	ArcM (78%)	0.109
VCD (81%)	ArcM (78%)	0.833

(e) Network 5

Vis. A (mean)	Vis. B (mean)	p-value
AM (71%)	VCD (73%)	0.895
AM (71%)	ArcM (78%)	0.583
VCD (73%)	ArcM (78%)	0.680

(f) Network 6

Table B.7: Difference in accuracy achieved on two different visualizations, respectively, across each network for DC task. Significant difference (according to Welch's t-test) highlighted in bold.

Vis. A (mean)	Vis. B (mean)	p-value
AM (8.33s)	VCD (11.58s)	0.041
AM (8.33s)	ArcM (11.04s)	0.079
VCD (11.58s)	ArcM (11.04s)	0.782

(a) Network 1

Vis. A (mean)	Vis. B (mean)	p-value
AM (8.30s)	VCD (9.73s)	0.372
AM (8.30s)	ArcM (15.47s)	0.007
VCD (9.73s)	ArcM (15.47s)	0.043

(b) Network 2

Vis. A (mean)	Vis. B (mean)	p-value
AM (11.72s)	VCD (12.39s)	0.770
AM (11.72s)	ArcM (16.64s)	0.036
VCD (12.39s)	ArcM (16.64s)	0.137

(c) Network 3

Vis. A (mean)	Vis. B (mean)	p-value
AM (10.93s)	VCD (12.12s)	0.537
AM (10.93s)	ArcM (15.82s)	0.028
VCD (12.12s)	ArcM (15.82s)	0.151

(d) Network 4

Vis. A (mean)	Vis. B (mean)	p-value
AM (18.54s)	VCD (26.07s)	0.103
AM (18.54s)	ArcM (15.06s)	0.215
VCD (26.07s)	ArcM (15.06s)	0.018

(e) Network 5

Vis. A (mean)	Vis. B (mean)	p-value
AM (14.60s)	VCD (19.88s)	0.205
AM (14.60s)	ArcM (24.90s)	0.024
VCD (19.88s)	ArcM (24.90s)	0.374

(f) Network 6

Table B.8: Difference in answer time recorded on two different visualizations, respectively, across each network for DC task. Significant difference (according to Welch's t-test) highlighted in bold.

B.5 Find Path (FP)

Vis. A (mean)	Vis. B (mean)	p-value
AM (43%)	VCD (42%)	0.968
AM (43%)	ArcM (52%)	0.517
VCD (42%)	ArcM (52%)	0.500

(a) Network 1

Vis. A (mean)	Vis. B (mean)	p-value
AM (32%)	VCD (46%)	0.301
AM (32%)	ArcM (52%)	0.157
VCD (46%)	ArcM (52%)	0.682

(b) Network 2

Vis. A (mean)	Vis. B (mean)	p-value
AM (29%)	VCD (31%)	0.863
AM (29%)	ArcM (52%)	0.093
VCD (31%)	ArcM (52%)	0.136

(c) Network 3

Vis. A (mean)	Vis. B (mean)	p-value
AM (25%)	VCD (27%)	0.875
AM (25%)	ArcM (26%)	0.931
VCD (27%)	ArcM (26%)	0.948

(d) Network 4

Vis. A (mean)	Vis. B (mean)	p-value
AM (25%)	VCD (31%)	0.645
AM (25%)	ArcM (30%)	0.675
VCD (31%)	ArcM (30%)	0.980

(e) Network 5

Vis. A (mean)	Vis. B (mean)	p-value
AM (36%)	VCD (19%)	0.180
AM (36%)	ArcM (17%)	0.141
VCD (19%)	ArcM (17%)	0.871

(f) Network 6

Table B.9: Difference in accuracy achieved on two different visualizations, respectively, across each network for FP task. Significant difference (according to Welch's t-test) highlighted in bold.

Vis. A (mean)	Vis. B (mean)	p-value
AM (15.10s)	VCD (25.39s)	0.038
AM (15.10s)	ArcM (22.63s)	0.036
VCD (25.39s)	ArcM (22.63s)	0.589

(a) Network 1

Vis. A (mean)	Vis. B (mean)	p-value
AM (27.94s)	VCD (20.09s)	0.170
AM (27.94s)	ArcM (30.71s)	0.621
VCD (20.09s)	ArcM (30.71s)	0.033

(b) Network 2

Vis. A (mean)	Vis. B (mean)	p-value
AM (23.76s)	VCD (20.72s)	0.466
AM (23.76s)	ArcM (50.24s)	0.006
VCD (20.72s)	ArcM (50.24s)	0.003

(c) Network 3

Vis. A (mean)	Vis. B (mean)	p-value
AM (26.50s)	VCD (29.65s)	0.627
AM (26.50s)	ArcM (49.26s)	0.009
VCD (29.65s)	ArcM (49.26s)	0.036

(d) Network 4

Vis. A (mean)	Vis. B (mean)	p-value
AM (39.46s)	VCD (36.32s)	0.716
AM (39.46s)	ArcM (53.91s)	0.209
VCD (36.32s)	ArcM (53.91s)	0.133

(e) Network 5

Vis. A (mean)	Vis. B (mean)	p-value
AM (50.05s)	VCD (33.61s)	0.165
AM (50.05s)	ArcM (43.61s)	0.617
VCD (33.61s)	ArcM (43.61s)	0.263

(f) Network 6

Table B.10: Difference in answer time recorded on two different visualizations, respectively, across each network for FP task. Significant difference (according to Welch's t-test) highlighted in bold.

C Demographics

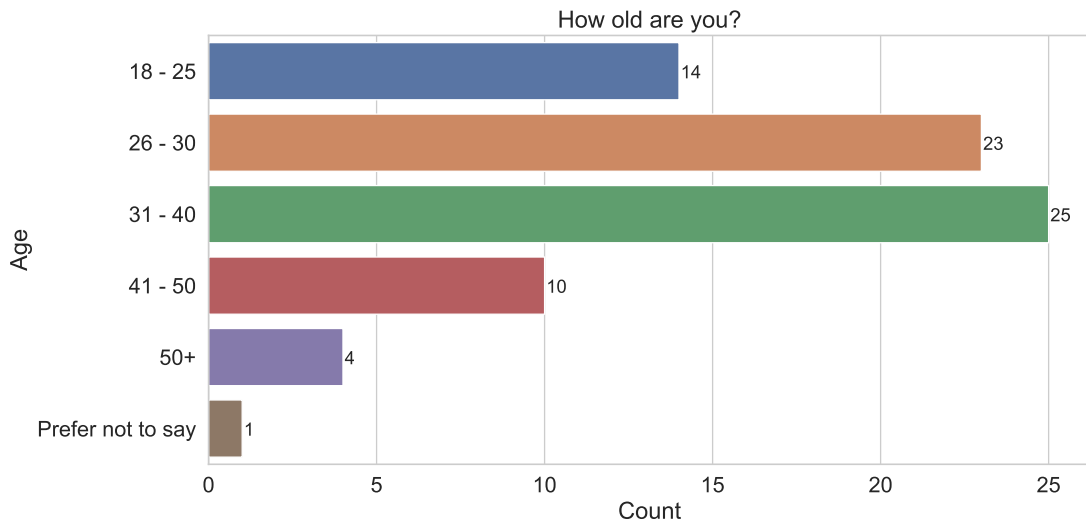


Figure C.1: Distribution of the participants based on their age.

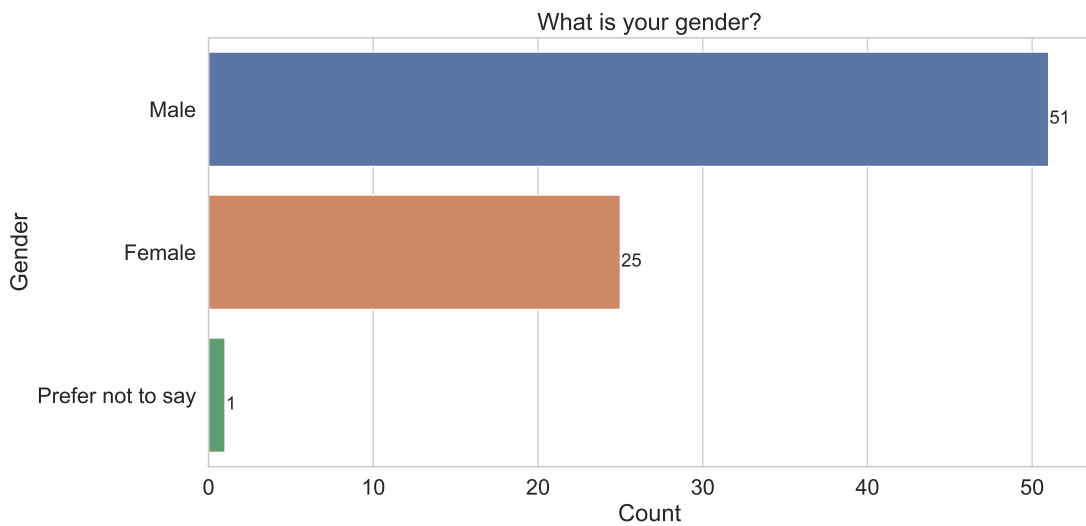


Figure C.2: Distribution of the participants based on their gender.

C Demographics

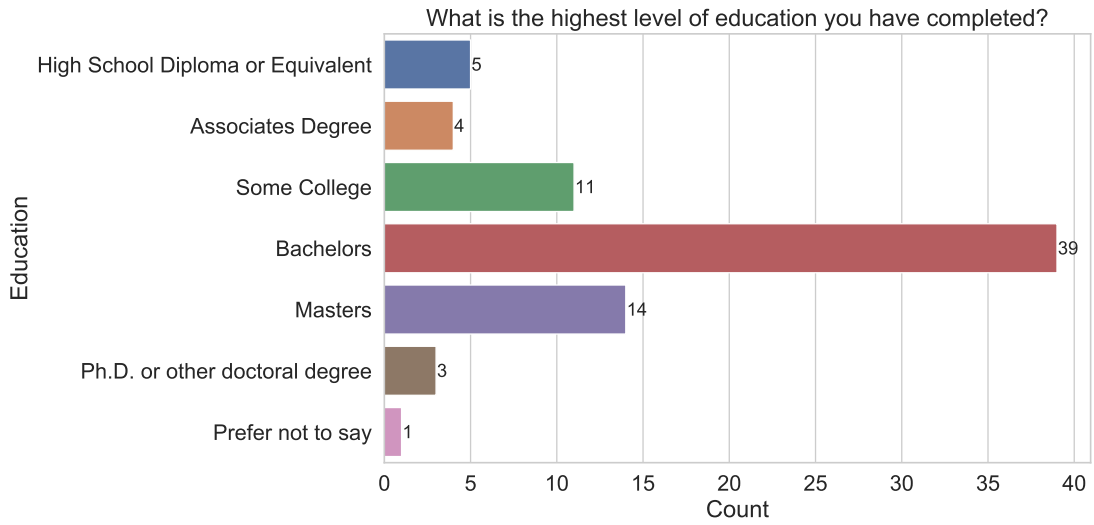


Figure C.3: Distribution of the participants based on their education.

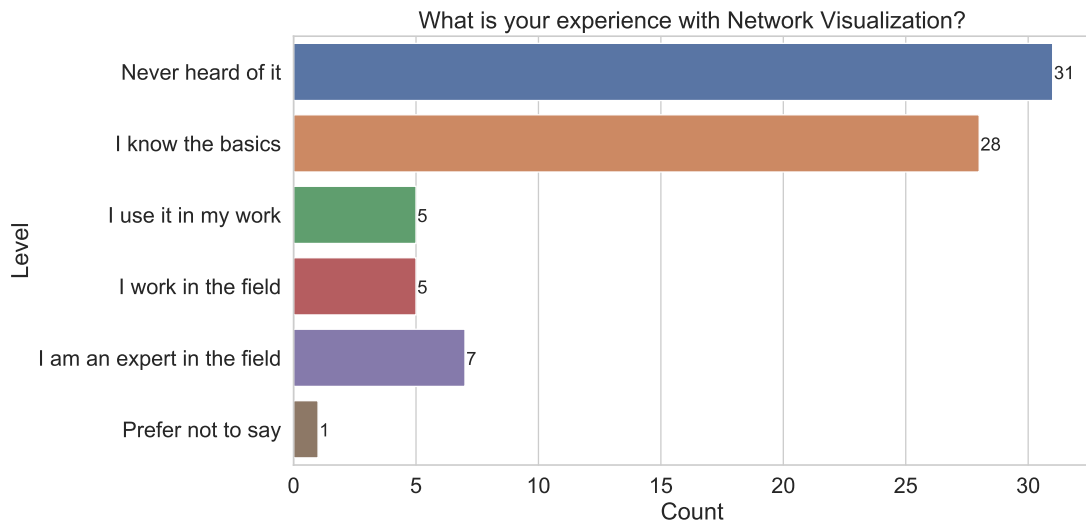


Figure C.4: Distribution of the participants based on their level of experience with network visualization.

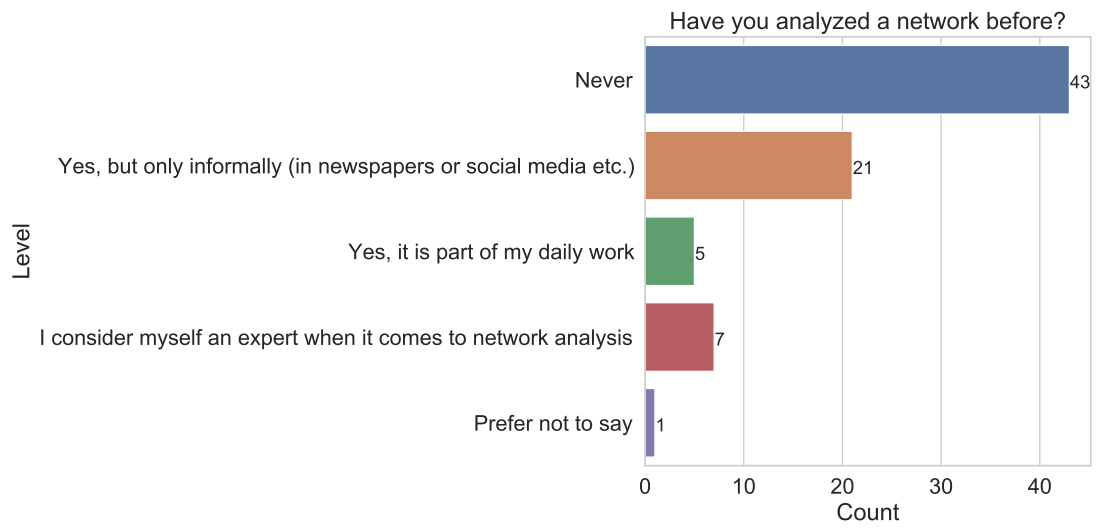


Figure C.5: Distribution of the participants based on their level of knowledge regarding network analysis.

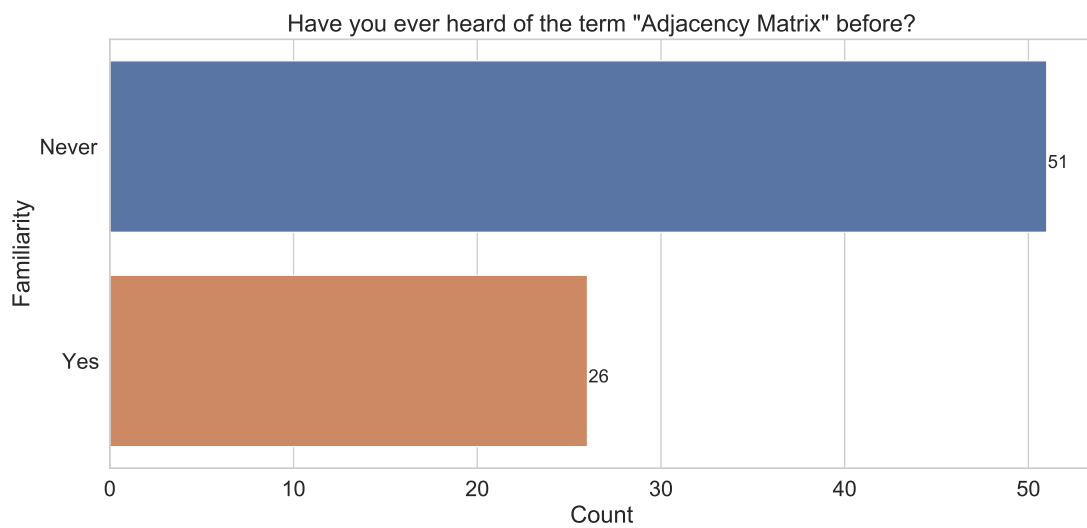


Figure C.6: Distribution of the participants based on their familiarity with the term “Adjacency Matrix”.

D Data concerning Vision Deficiency

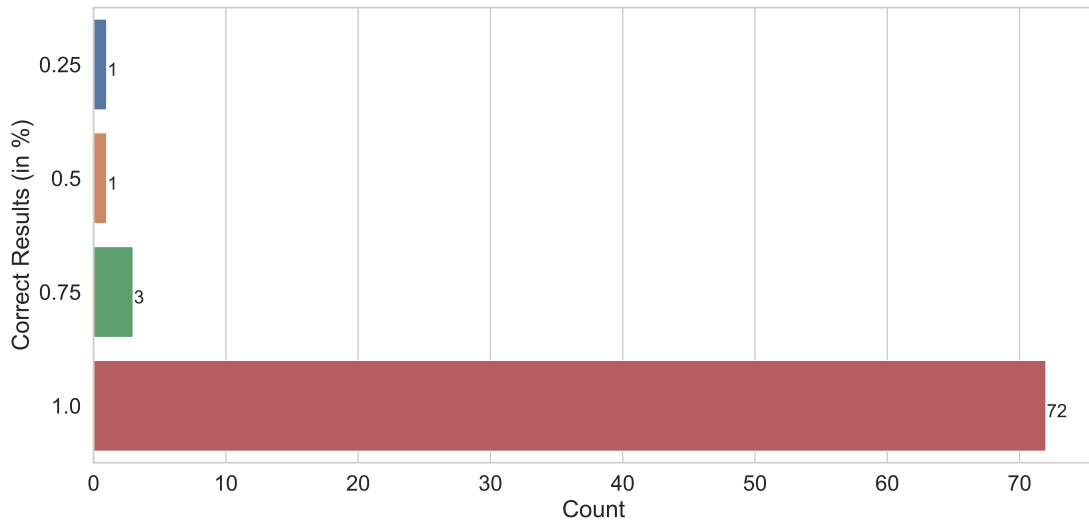


Figure D.1: Distribution of the participants based on their performance in the color-blindness test.

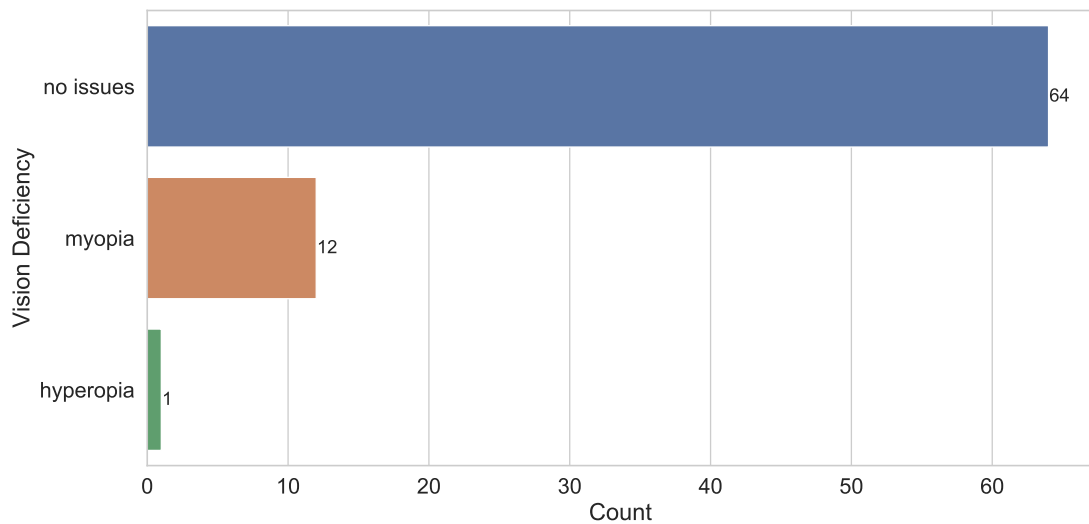


Figure D.2: Distribution of the participants based on vision deficiencies they potentially suffer from.

Bibliography

- [AB02] R. Albert, A.L. Barabási. “Statistical Mechanics of Complex Networks”. In: *Reviews of Modern Physics* 74 (2002), pp. 47–97. ISSN: 00346861. DOI: [10.1103/RevModPhys.74.47](https://doi.org/10.1103/RevModPhys.74.47). arXiv: [0106096](https://arxiv.org/abs/0106096) [cond-mat] (cit. on pp. 15, 44).
- [BB04] M. Baur, U. Brandes. “Crossing Reduction in Circular Layouts”. In: *Hromkovič J., Nagl M., Westfechtel B. (eds) Graph-Theoretic Concepts in Computer Science. WG 2004. Lecture Notes in Computer Science* (2004), pp. 332–343. ISSN: 16113349. DOI: [10.1007/978-3-540-30559-0_28](https://doi.org/10.1007/978-3-540-30559-0_28) (cit. on p. 16).
- [BR79] H. B. Barlow, B. C. Reeves. *The versatility and absolute efficiency of detecting mirror symmetry in random dot displays*. 1979. DOI: [10.1016/0042-6989\(79\)90154-8](https://doi.org/10.1016/0042-6989(79)90154-8) (cit. on pp. 31, 34).
- [BRSG07] C. Bennett, J. Ryall, L. Spalteholz, A. Gooch. “The Aesthetics of Graph Visualization”. In: *Proceedings of the 2007 Computational Aesthetics in Graphics, Visualization, and Imaging* (2007), pp. 57–64. DOI: [10.2312/COMPAESTH/COMPAESTH07/057-064](https://doi.org/10.2312/COMPAESTH/COMPAESTH07/057-064) (cit. on p. 16).
- [dF10] J. de Las Rivas, C. Fontanillo. “Protein-Protein Interactions Essentials: Key Concepts to Building and Analyzing Interactome Networks”. In: *PLoS Computational Biology* 6.6 (2010), pp. 1–8. ISSN: 1553734X. DOI: [10.1371/journal.pcbi.1000807](https://doi.org/10.1371/journal.pcbi.1000807) (cit. on p. 15).
- [DW16] B. Derrick, P. White. “Why Welch’s test is Type I error robust”. In: *The Quantitative Methods for Psychology* 12.1 (2016), pp. 30–38. DOI: [10.20982/tqmp.12.1.p030](https://doi.org/10.20982/tqmp.12.1.p030) (cit. on p. 51).
- [FARZ18] J. Fagnan, A. Abnar, R. Rabbany, O. R. Zaïane. “Modular Networks for Validating Community Detection Algorithms”. In: (2018). ISSN: 23318422. arXiv: [1801.01229](https://arxiv.org/abs/1801.01229) (cit. on pp. 43–45).
- [FK13] J. J. Fowler, S. G. Kobourov. “Planar preprocessing for spring embedders”. In: *Didimo W., Patrignani M. (eds) Graph Drawing. GD 2012. Lecture Notes in Computer Science LNCS 7704* (2013), pp. 388–399. ISSN: 03029743. DOI: [10.1007/978-3-642-36763-2_35](https://doi.org/10.1007/978-3-642-36763-2_35) (cit. on p. 16).
- [GFC04] M. Ghoniem, J. D. Fekete, P. Castagliola. “A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations”. In: *Proceedings - IEEE Symposium on Information Visualization, INFO VIS* (2004), pp. 17–24. ISSN: 1522404X. DOI: [10.1109/INFVIS.2004.1](https://doi.org/10.1109/INFVIS.2004.1) (cit. on pp. 17, 19, 24).
- [HF06] N. Henry, J. D. Fekete. “MatrixExplorer: a Dual-Representation System to Explore Social Networks”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.5 (2006), pp. 677–684. ISSN: 10772626. DOI: [10.1109/TVCG.2006.160](https://doi.org/10.1109/TVCG.2006.160) (cit. on p. 22).

- [HF07] N. Henry, J. D. Fekete. “MatLink: Enhanced Matrix Visualization for Analyzing Social Networks”. In: *Proceedings of IFIP TC13 International Conference on Human-Computer Interaction (INTERACT)* (2007), pp. 288–302. ISSN: 16113349. DOI: [10.1007/978-3-540-74800-7_24](https://doi.org/10.1007/978-3-540-74800-7_24) (cit. on pp. 17, 20–22, 70).
- [HFM07] N. Henry, J. D. Fekete, M. J. McGuffin. “NodeTriX: A Hybrid Visualization of Social Networks”. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1302–1309. ISSN: 10772626. DOI: [10.1109/TVCG.2007.70582](https://doi.org/10.1109/TVCG.2007.70582) (cit. on pp. 17, 21, 22, 70).
- [Kad12] C. Kadushin. *Understanding Social Networks: Theories, Concepts, And Findings*. Oxford University Press, 2012, p. 251. ISBN: 9780195379471 (cit. on p. 15).
- [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 5.1* (2006), pp. 62–76. ISSN: 14738716. DOI: [10.1057/palgrave.ivs.9500116](https://doi.org/10.1057/palgrave.ivs.9500116) (cit. on pp. 16, 17, 19, 24).
- [LPP+06] B. Lee, C. Plaisant, C. Parr, J.-D. Fekete, N. Henry Riche. “Task Taxonomy for Graph Visualization”. In: *Proceedings of AVI Workshop on BEyond time and Errors* (2006), pp. 1–5. DOI: [10.1145/1168149.1168168](https://doi.org/10.1145/1168149.1168168) (cit. on p. 16).
- [MB09] D. K. Michael, S. Battiston. “From Graph Theory to Models of Economic Networks. A Tutorial”. In: *Lecture Notes in Economics and Mathematical Systems* (2009), pp. 23–63. ISSN: 00758442. DOI: [10.1007/978-3-540-68409-1_2](https://doi.org/10.1007/978-3-540-68409-1_2) (cit. on p. 15).
- [MD18] X. Ma, S. Dhavala. “Hierarchical Clustering with Prior Knowledge”. In: *arXiv* (2018). ISSN: 23318422. arXiv: [1806.03432](https://arxiv.org/abs/1806.03432) (cit. on pp. 46, 47).
- [New03] M. E. Newman. “Mixing patterns in networks”. In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 67.2 (2003), p. 13. ISSN: 1063651X. DOI: [10.1103/PhysRevE.67.026126](https://doi.org/10.1103/PhysRevE.67.026126). arXiv: [0209450](https://arxiv.org/abs/0209450) [cond-mat] (cit. on p. 45).
- [NWHL20] C. Nobre, D. Wootton, L. Harrison, A. Lex. “Evaluating Multivariate Network Visualization Techniques Using a Validated Design and Crowdsourcing Approach”. In: *Conference on Human Factors in Computing Systems - Proceedings* (2020), pp. 1–12. DOI: [10.1145/3313831.3376381](https://doi.org/10.1145/3313831.3376381) (cit. on pp. 19, 20).
- [OJK18] M. Okoe, R. Jianu, S. Kobourov. “Revisited Experimental Comparison of Node-Link and Matrix Representations”. In: *Graph Drawing and Network Visualization* (2018), pp. 287–302. ISSN: 16113349. DOI: [10.1007/978-3-319-73915-1_23](https://doi.org/10.1007/978-3-319-73915-1_23). arXiv: [1709.00293](https://arxiv.org/abs/1709.00293) (cit. on pp. 17, 19, 24).
- [RLMJ05] R. Rosenholtz, Y. Li, J. Mansfield, Z. Jin. “Feature Congestion: A Measure of Display Clutter”. In: *Proceedings of the SIGCHI* (2005), pp. 761–770. URL: <http://dl.acm.org/citation.cfm?id=1055078> (cit. on p. 16).
- [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* 7.2 (2019), pp. 242–264. ISSN: 20501250. DOI: [10.1017/nws.2019.6](https://doi.org/10.1017/nws.2019.6) (cit. on pp. 17, 20, 24).

- [RVN20] N. Randriamihamison, N. Vialaneix, P. Neuvial. “Applicability and Interpretability of Ward’s Hierarchical Agglomerative Clustering With or Without Contiguity Constraints”. In: *Journal of Classification* (2020). ISSN: 14321343. DOI: [10.1007/s00357-020-09377-y](https://doi.org/10.1007/s00357-020-09377-y) (cit. on p. 47).
- [SBPP15] J. Sansen, R. Bourqui, B. Pinaud, H. Purchase. “Edge Visual Encodings in Matrix-Based Diagrams”. In: *19th International Conference* (2015), pp. 62–67. DOI: [10.1109/iv.2015.22](https://doi.org/10.1109/iv.2015.22) (cit. on pp. 17, 21, 29).
- [Wag95] J. Wagemans. “Detection of visual symmetries”. In: *Spatial Vision* 9.1 (1995), pp. 9–32. ISSN: 0169-1015. DOI: [10.1163/156856895X00098](https://doi.org/10.1163/156856895X00098) (cit. on pp. 31, 34).
- [Wat02] M. Wattenberg. “Arc diagrams: Visualizing Structure in Strings”. In: *Proceedings - IEEE Symposium on Information Visualization, INFO VIS* (2002), pp. 110–116. ISSN: 1522404X. DOI: [10.1109/INFVIS.2002.1173155](https://doi.org/10.1109/INFVIS.2002.1173155) (cit. on p. 29).
- [WS98] D. J. Watts, S. H. Strogatz. “Collective dynamics of ‘small-world’ networks”. In: *Nature* 393 (1998), pp. 440–442. DOI: [10.1038/30918](https://doi.org/10.1038/30918) (cit. on p. 45).

All links were last followed on April 23, 2021.

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