GraphVisual: Design and Evaluation of a Web-Based Visualization Tool for Teaching and Learning Graph Visualization

Martin Imre[1], Wenqing Chang[2], Shuzhan Wang[3], Christine Trinter[1], Chaoli Wang[1]
[1]University of Notre Dame, [2]Xi'an Jiaotong University, [3]Beijing University of Posts and Telecommunications

Abstract

Graphs are ubiquitous in representing data from various fields, such as biological interactions, geographic knowledge, and engineering problems. They allow users to acquire a quick overview of how the data entities are spread and connected in order to gain insights from the data. Many software tools exist to depict and explore graphs. However, there is a lack of tools dedicated to assisting students in learning graph visualization and related concepts. We present GraphVisual, an educational software tool that provides an accessible web interface for students to interact with and explore graph visualizations of real-world data sets of different sizes and characteristics. GraphVisual integrates several popular graph layout algorithms to enable users to discover patterns, outliers, and other features of these data sets. GraphVisual allows users to compare and evaluate different graph visualization techniques from two side-by-side graph display panels, which are supported by a set of interaction functions. To demonstrate the utility of GraphVisual and assess its effectiveness, we conducted a formal user study with two groups: students who were enrolled in a college-level data visualization course and doctoral computer science students who did not take the course. The study includes an introduction, a training session, a survey, and an in-class quiz that is integrated into GraphVisual.

1 Introduction

Graphs and networks are an omnipresent concept to explain relationships in data. For example, graphs can be used to model biological data such as neurons and the interactions among them, geographic data like airports and the flight connections between them, or mathematical problems such as the shortest path problem. However, to generate those insights in a human-understandable way, the graph needs to be visualized. Graph visualization is well researched, and there exist tools to visualize graphs of any kind, yet challenges remain. A key challenge is to pick a suitable layout for the data that not only represents the underlying graph well but also does not overwhelm the viewers. To overcome such challenges to effective graph visualization, we need to understand graph visualization better. In most cases, graph visualization is taught to students by showing them different graph drawings and explaining the fundamental concepts of layout algorithms theoretically. However, it is often difficult to interact, compare, and explore the multitude of options when it comes to visualizing a graph. Therefore, developing a pedagogical visualization software tool has the potential to amplify the students' understanding of graph visualization concepts.

In this paper, we present GraphVisual, a web-based educational tool for instructors to teach and students to learn graph visualizations using real-world data sets. A variety of graph layout algorithms, including animated, fully interactive physics-based simulations, allow users to discover patterns, clusters, and other features of these data sets, which are drawn from a broad spectrum of fields, including biology, geography, and social networks. GraphVisual allows users to learn how to compare and evaluate different graph visualizations in two side-by-side display panels, which can be tweaked, reformed, and augmented by various interactions. For example, selection and filtering functions make even very dense graphs easy to visualize and understand.

Our core deliverable is a web-based interface that allows instructors to demonstrate graph layouts in class, and students to further self-study graph visualization after class. GraphVisual has been used in a classroom environment, and its effectiveness was evaluated through a formal user study involving students from computer science and engineering, applied and computational mathematics and statistics, and aerospace and mechanical engineering.

2 Related Work

Graph drawing or visualization is an essential part of structured data analysis and has been studied thoroughly. Graphs are often visualized by node-link diagrams with nodes as entities and links between them as connections. The central aspect of graph visualization is layout generation, which assigns every node a position on a 2D-plane for drawing. The main challenges to graph visualization are scalability, readability, and aesthetics. For a detailed overview of graph visualization and its applications, we refer interested readers to the following surveys [16, 11, 15, 4, 24]. In GraphVisual, we categorize graph layouts into force-directed and non-force-directed layouts. Force-directed layouts use a physics-based simulation that assigns attractive and repulsive forces between pairs of nodes, aiming to minimize the energy function of these forces. GraphVisual supports the D3-force-directed layout [7] which is an implementation and adaptation of Dwyer's [12] algorithm, fast multipole multilevel method (FM^3) [14], ForceAtlas2 [20] which focuses on network spatialization, OpenOrd [22] which is designed for large graphs, and Yifan Hu's algorithm [18] which combines a multilevel approach, together with Barnes and Hut octree force calculation [2] and an adaptive cooling scheme for the general force model for efficient high-quality graph drawing. Among non-force-directed layouts, GraphVisual supports a spectral layout which uses the spectrum of the graph to assign node positions [19], a circular layout which simply aligns nodes along a circle, and a random layout which assigns node positions randomly. In addition, GraphVisual supports two special layouts, geographical layout and bipartite layout, for drawing graphs with geographical information and bipartite graphs, respectively.

Teaching information visualization, including graph visualization, has not changed significantly over the years. Back in 2005, "Education and Training" was listed as one of the top ten unsolved problems in information visualization [9]. The overarching theme for the 2015 Gordon Research Conference on Visualization in Science and Education is "Grand Challenges in the Use of Visualization in Science and Education" [1]. More recently, educational psychology concepts were applied to design four types of online guides for information visualization [23]. However, to the best of our knowledge, few works focus on the pedagogical aspects of graph visualization.

There exist many tools that focus on visualizing graphs, for example, Gephi [3] and Graphviz [13], just to name a few. These tools, however, provide users with a rich set of functions along with an abundance of options, making it difficult for beginner students to learn the critical aspects of graph visualization quickly. GraphVisual overcomes this issue by only offering the most important functions and options to users. Other tools use graph visualization to teach graph algorithms or other concepts. For example, CABRI [8], LINK [5], and EVEGA [21] were introduced to teach graph algorithms such as breadth-first search (BFS) or depth-first search (DFS). These tools commonly allow for dynamic graph construction and visualization and mainly focus on stepwise execution of a given algorithm. Another interesting work by Cohen et al. [10] aims at teaching basic graph visualization concepts to visually impaired students. They used different audio encodings for nodes and edges to help visually impaired students navigate a graph drawing on a tablet. However, none of these approaches focuses on supporting graph visualization education, which is the primary focus of GraphVisual. With a clear interface, GraphVisual provides the essential components for instructors to teach and students to learn the core concepts and algorithms of graph visualization.

3 Design Principles

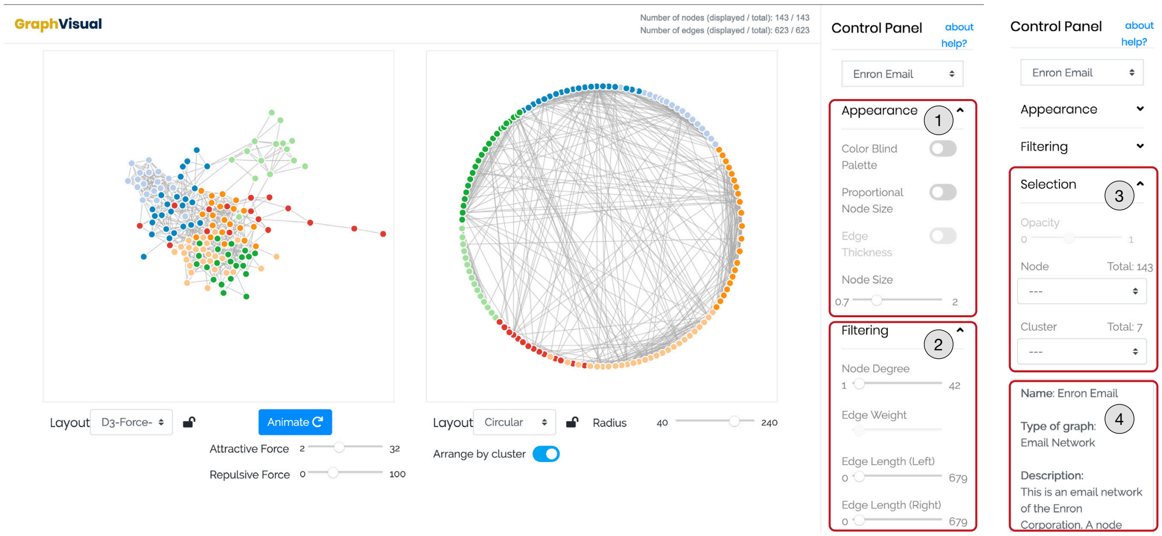
Designing an excellent user interface must target the end users from the beginning and follow a set of carefully thought principles. For GraphVisual, we aim to not only present a concise, portable, and easy-to-use interface but also retain it as a learning tool for beginner students rather than for experts of graph visualization. We, therefore, designed GraphVisual with the following principles in mind.

P1: Simplicity. The most important principle for GraphVisual is to make it simple to use. There are ample tools available that allow users to visualize various graph data and tweak a multitude of parameters. However, these tools are often overwhelming for beginner users, and thus would also be unfit for students learning graph visualization. Therefore, we aim to keep the user interface simple, self-explanatory, and with a limited set of functions and parameters.

P2: Selectivity. Graph visualization is a fairly broad topic: different layout algorithms, graph data sets, and interaction functions could quickly overwhelm users. As an educational tool, the simplicity of GraphVisual implies that we should be selective in terms of deciding which layout algorithms, data sets, and interaction functions to include. For instance, in addition to straightforward forms of random and circular graph layouts as the baseline, we also selectively include non-trivial ones, such as force-directed layouts and other layouts (spectral, geographical, and bipartite). The same principle is applied to choose representative data sets (from varying sizes, characteristics, and domains) and essential interaction functions.

P3: Interactivity. Interactivity is the key to data visualization, which is no exception to any educational tool for learning data visualization. Therefore, a critical aspect of engaging students in learning GraphVisual is to enable interactions. "Learning by doing" is a common concept in pedagogical design. As such, we employ an essential set of interactions for GraphVisual. GraphVisual provides functions to support filtering and selection functions in conjunction with different data sets under several graph layouts. To further deepen the learning effect, we allow adjusting parameters for a subset of the layouts so that users can gain instantaneous feedback.

P4: Comparison. When studying multiple related concepts, it is often helpful to compare their respective strengths and weaknesses. For graph visualization, the primary choice lies in the selection of a particular layout to draw a given graph. GraphVisual supports simultaneous visualization of two different graph layouts of the same data set in two side-by-side display panels, allowing students to make easy comparison through brushing and linking (i.e., the selection made in one view is dynamically reflected in the other view). Furthermore, the two display panels react synchronically to interactions such as filtering and selection of nodes and edges.



Name: Enron Email

Type of graph: Email Network

Description: This is an email network of the Enron Corporation. A node

(a) (b)

Figure 1: The user interface of GraphVisual (study component). (a) The entire screenshot with Appearance and Filtering unfolded on the control panel. (b) The remainder of the control panel showing Selection and data set information.

4 Design and Implementation

We developed GraphVisual to assist students in learning graph visualization and related concepts. Users can explore graph visualizations of several real-world data sets. Our tool includes several kinds of interaction functions, including switching data sets, filtering nodes and edges, and selecting nodes or clusters. To change a graph's appearance, we offer options such as "Color Blind Palette," "Proportional Node Size," and "Edge Thickness" to provide users with more flexibility in exploring the graph. GraphVisual has two main components: the *study component* and *quiz component*.

Our visualization interface is implemented using HTML, CSS, and JavaScript, especially using D3.js. Figure 1 shows a screenshot of the interface with three major parts: a header bar on the top, a central panel in the middle, and a scrollable control panel on the right. The top bar displays the GraphVisual logo and node and edge information of the graph. The central panel is composed of two side-by-side display panels, each having a graph drawing and a set of control widgets (e.g., a dropdown list for layout selection) for the corresponding graph drawing. The control panel provides a variety of interaction functions that work on both drawings. On the top, there is a dropdown list for data set selection, and two links "about" and "help?" for reviewing the tool's onboarding information and the "about" page information, respectively. Beneath that, there are four main sections: the Appearance (1), Filtering (2), Selection (3), and data description (4) sections. Each of the first three sections can be collapsed or expanded to show interaction options as needed. GraphVisual also shows tooltips next to various elements when hovering them, while the most important tooltips are displayed at the bottom of the central panel, introducing the primary interactions that can be performed on graphs. In the following, we describe the real-world data sets, graph layouts, filtering and selection functions, and the difference between the study and quiz components.

Name	Domain
Enzymes	Biological
Mouse Visual Cortex	Biological
Cities and Firms	Bipartide
Sandi Authors	Collaboration
Airfoil	Engineering
Least Squares	Engineering
Robot	Engineering
Small Generalized Eigenvalue	Engineering
Football	Geographical
Les Miserables	Social
Relation	Social
Retweet	Social
Power System	Structural
Transmission Tower	Structural

Name	Domain
Cat Brain	Biological
Divorce	Bipartide
Netscience	Collaboration
USAir97	Geographical
Dolphins	Social
Enron Email	Social
Naval	Structural

Table 1: The data sets included in GraphVisual. The left and right tables show the data sets included in the study and quiz components, respectively.

4.1 Data Sets

Graph visualization has been widely used in social and biological sciences to visualize structural networks. Intended for STEM students to learn data visualization, GraphVisual includes selective data sets from these and other diverse domains. Table 1 shows the data sets we include in GraphVisual. The left and right tables list the data sets included in the study and quiz components, respectively. We ensure that every data domain used in the study component is also represented in the quiz component. Based on the numbers of nodes and edges, these data sets are categorized into small, medium, and large graphs on the interface.

4.2 Graph Layouts

GraphVisual supports different graph layouts for users to discover structures, patterns, and other features of the data sets. These layouts are divided into two categories: force-directed and non-force-directed layouts. As a popular approach for graph drawing, force-directed layout algorithms aim to draw graphs that are aesthetically pleasing and easy to interpret. The layout is achieved by assigning attractive and repulsive forces between pairs of nodes and then using these forces to minimize an energy function. Non-force-directed algorithms use other strategies to compute the positions of nodes and generate reasonable visualizations. By default, we provide the options of circular, random, and spectral layouts in the non-force-directed category for each data set. We also include two additional non-force-directed layouts, geographical and bipartite layouts. They are used for drawing data sets that have geographic information and exhibit bipartite properties.

Force-directed Layouts. GraphVisual allows users to adjust attractive and repulsive forces to explore their impact on the D3-force-directed layout [7]. Students can learn about force-directed algorithms better by adjusting forces and observing the graph layout changes. Figure 2 shows three D3-force-directed graph drawings of the Les Miserables data set with different combinations of attractive and repulsive forces. The first drawing (a) shows our default setting with the attractive force of 5 and a repulsive force of 30. The second one (b) shows the graph with a higher attractive force and a lower repulsive force, presenting a more contracted drawing than the first. The third one (c) shows the drawing with a lower attractive force and a higher repulsive force, presenting a more expanded drawing than the first. Users can trigger a slower re-computation of the layout by clicking the “Animate” button, which allows them to understand better how the forces are updated iteratively.

In addition to D3-force-directed layout, GraphVisual supports four other popular force-directed layouts: FM^3, ForceAtlas2, Yifan Hu, and OpenOrd layouts (refer to Section 2). Figure 3 shows the examples of each one. In this figure, (a)

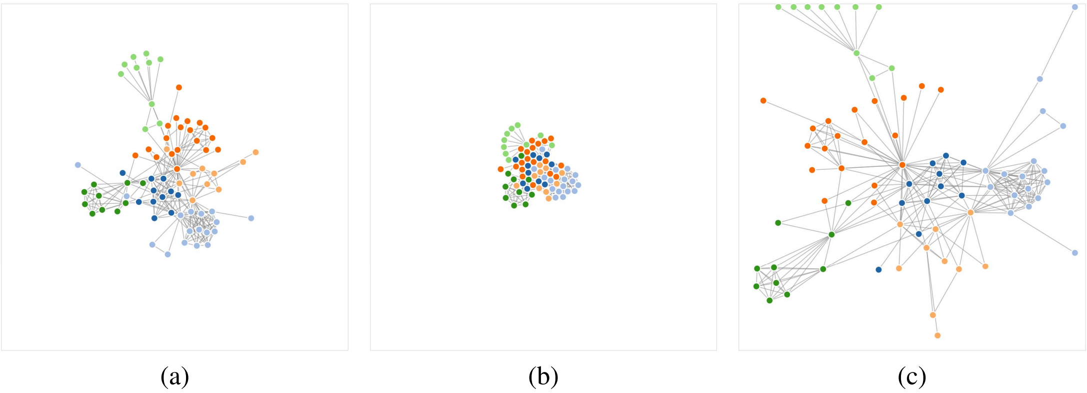


Figure 2: Three different D3-force-directed graph drawings of the Les Miserables data set with different combinations of attractive and repulsive forces. (a) Default: attractive force: 5, repulsive force: 30; (b) Contracted: attractive force: 25, repulsive force: 2.5; (c) Expanded: attractive force: 3, repulsive force: 100.

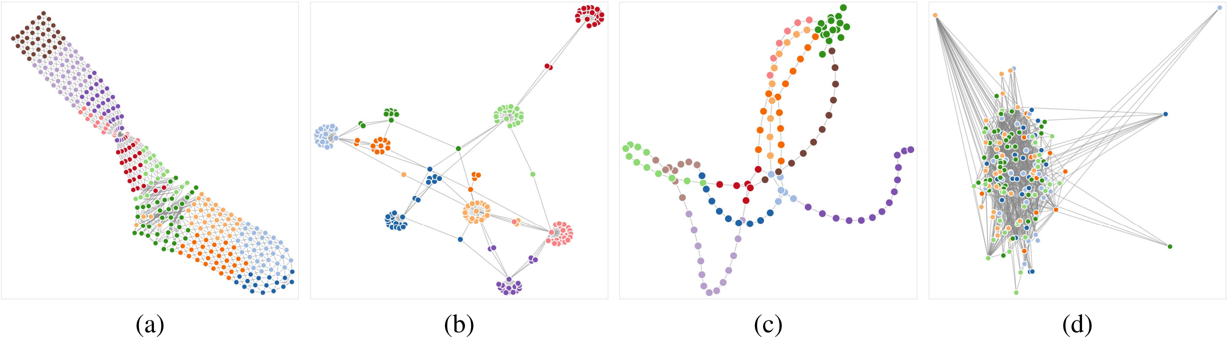


Figure 3: Graph drawings using force-directed layouts. (a) FM^3 layout for the Naval data set, (b) ForceAtlas2 layout for the Mouse Visual Cortex data set, (c) Yifan Hu's layout for the Enzymes data set, and (d) OpenOrd layout for the Relation data set.

shows the FM^3 graph drawing of the Naval data set, (b) is the ForceAtlas2 graph drawing of the Mouse Visual Cortex data set, (c) renders the Yifan Hu layout for the Enzymes data set, and (d) displays the OpenOrd graph layout for the Relation data set.

Non-force-directed Layouts. To enable the comparison among force-directed and non-force-directed layouts, GraphVisual supports a set of non-force-directed graph layouts. Specifically, we include three layouts (i.e., random, circular, and spectral) for all graphs, and two data-specific layouts (geographical and bipartite). The random and circular layouts are trivial and have minimal interaction (e.g., re-randomization, radius adjustment). Thus, we only discuss spectral, geographical, and bipartite layouts in the following paragraphs.

Spectral Layout. The spectral layout uses the underlying information about the graph's spectrum to position the nodes based on the eigenvectors of the graph's Laplacian matrix. We use the first two dominant eigenvectors as node positions for the spectral layout. Due to its underlying spectral method, this strength of this layout lies in visualizing structured data. Here we compare two graph drawings with the spectral layout. Figure 4 (a) visualizes Airfoil, a structural data set of a sparse representation mesh graph from finite element analysis. We can see that the airfoil structure is well preserved, showing the advantage of using the spectral layout in the visualization. By contrast, Figure 4 (b) shows the visualization of Cat Brain, a non-structural data set. Although we can see outliers easily from the spectral layout, the layout itself does not reveal much insight about the rest of the graph.

Geographical Layout. GraphVisual supports a special layout for geographical data sets. In this layout, nodes are positioned to reflect their real-world geographical locations. For example, the USAir97 data set is a record of the US air flights in 1997, in which a node represents an airport and edges are flight connections between airports. In Figure 4 (c), it is clear that the node (airport) positions resemble the geographical shape of the United States. Such a layout

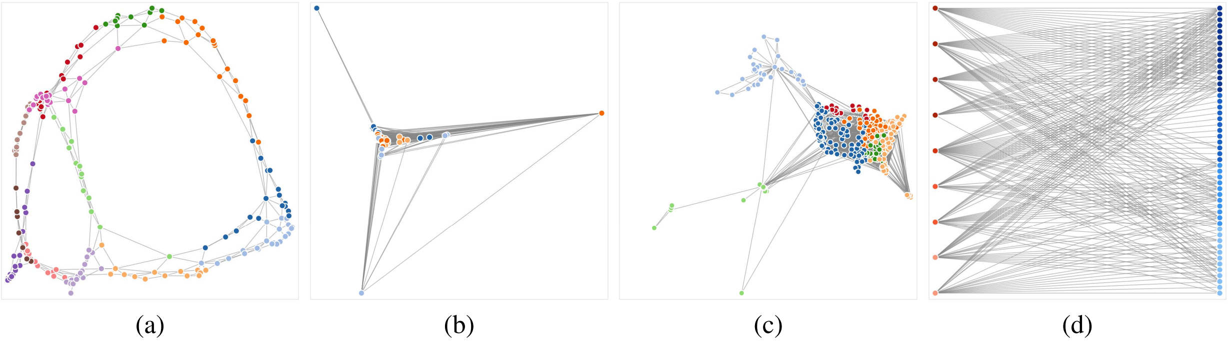


Figure 4: Non-force-directed graph layouts. (a) and (b) The spectral layouts for the Airfoil and Cat Brain data sets. (c) The geographical layout for the USAir97 data set. (d) The bipartite layout for the Divorce data set.

allows viewers to connect the graph with prior knowledge for a better understanding.

Bipartite Layout. A bipartite graph is a graph whose nodes can be divided into two disjoint and independent sets **U** and **V** such that edges only connect nodes in **U** to nodes in **V**. For these data sets, GraphVisual supports the bipartite layout. Figure 4 (d) shows the bipartite graph drawing of the Divorce data set, in which the set of nodes on the right represents the 50 states of the United States and the set of nodes on the left represents legal grounds for divorce. Such a layout allows users to distinguish the two different node sets better than other layouts.

4.3 Interaction Functions

GraphVisual provides filtering and selection functions for users to explore and understand the underlying graphs, supporting them to investigate nodes of interest with a less clutter view and answer questions about the graphs.

Filtering. Filtering allows users to filter graph nodes or edges by different criteria for easier visualization of large graphs or a better understanding of specific information. GraphVisual supports three different filters: node degree, edge weight, and edge length (in pixel unit) filters. As shown in Figure 1 (b, 2), we provide three kinds of sliders for specific types of filters. Users can easily adjust a slider's value to set the threshold for the corresponding filter.

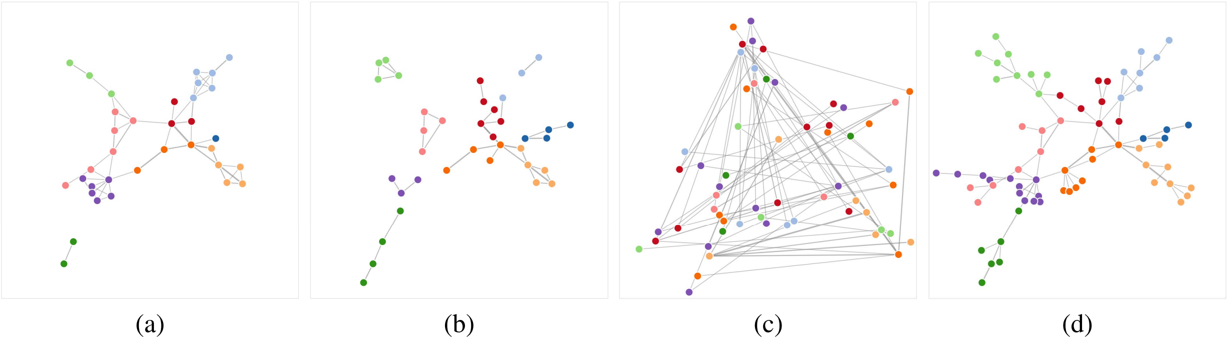


Figure 5: Filtering the Sandi Authors data set. (a) Filtered by node degree in the D3-force-directed layout. (b) Filtered by edge weight in the D3-force-directed layout. (c) Filtered by edge length in the FM^3 layout. (d) Filtered by edge length in the D3-force-directed layout.

Node degree indicates the number of edges adjacent to a node. Filtering by node degree allows users to remove low-degree nodes, which are often unimportant, from the graph drawing. Figure 5 (a) shows the nodes of the Sandi Authors data set in the D3-force-directed layout whose node degrees are 3 or above.

Edge weight represents different meanings, depending on the data set. For example, in the Sandi Authors data set, it shows the number of times two scientists collaborate. As this is data-specific, GraphVisual only supports this filter if the graph carries edge weights. Otherwise, this filter is disabled. Figure 5 (b) only shows the edges (and incident nodes) with a weight of 2 or above.

The edge length filters depend on the drawing of the graphs in different layouts. The length is specified in pixels. This filter uses either display panel as the base for determining which edges to remove, and then removing the same set of edges from both drawings. The two sliders, "Edge Length (Left)" and "Edge Length (Right)," allow users to filter the graphs based on the left and right drawings, respectively, and to compare the difference between the two given graph layouts. As shown in Figure 5 (c and d), the nodes and edges are filtered by the edge length in the FM^3 layout with a threshold value of 219 pixels on the left display panel (c), and the D3-force-directed layout with a threshold value of 34 pixels on the right display panel (d), respectively.

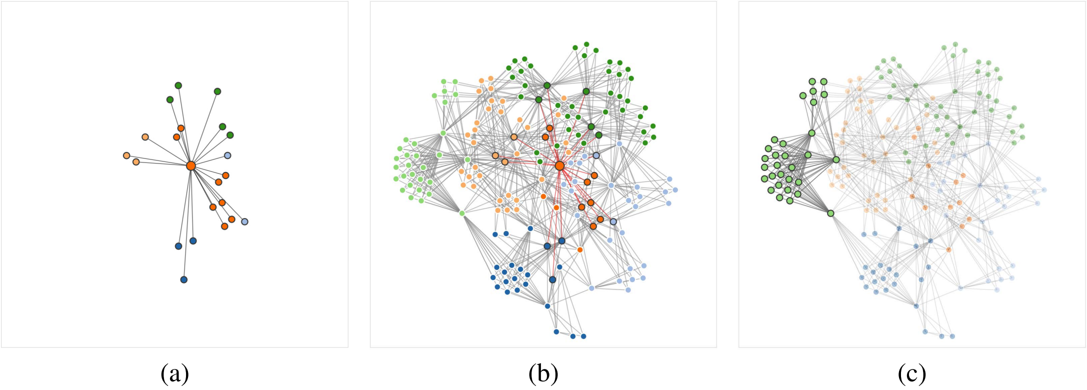


Figure 6: Node and cluster selection and highlighting of the Relation data set with different background opacities. (a) and (b) The selected node's one-ring neighborhood is highlighted with background opacity of 0.0 and 1.0, respectively. (c) The selected cluster is highlighted with background opacity of 0.4.

Selection. The selection functions allow users to select a node or a cluster for highlighting. Both nodes and clusters can be selected from the dropdown lists under the Selection section of the control panel, and nodes can also be selected by directly clicking on them on either of the display panels. The node clicked is highlighted with its one-ring neighborhood, as shown in Figure 6 (a). With a double-click of a node, its two-ring neighborhood is highlighted. When a node is highlighted, and a user hovers the mouse over it, the node's information (e.g., name and degree) is displayed in the tooltip.

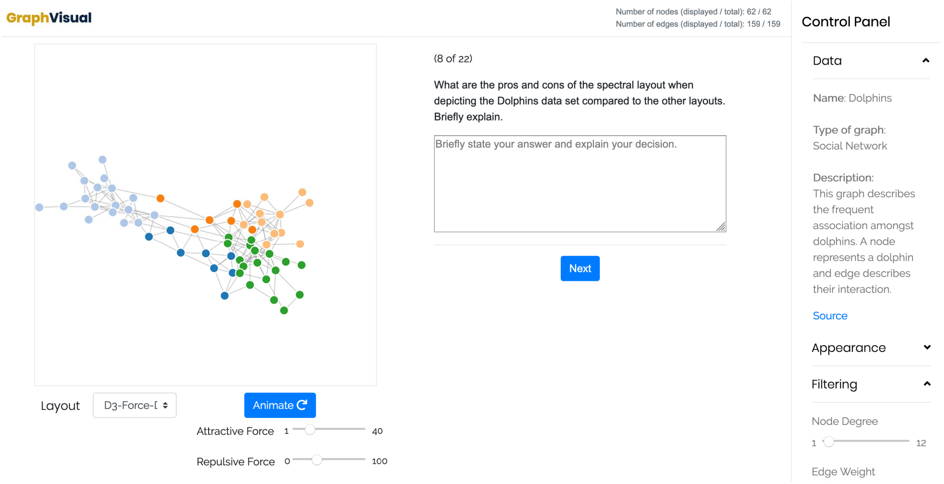
For clustering, GraphVisual uses the Louvain community detection algorithm [6]. Nodes within a cluster share the same color in the drawing. When a cluster is selected, all the nodes within the cluster are highlighted. Such an example is shown in Figure 6 (c).

To help users keep the context during highlighting, GraphVisual allows them to adjust the opacity (which is a value between 0 and 1) of the unselected nodes. Such effects can be seen in Figure 6 (a), (b), and (c), which have a background opacity of 0.0, 1.0, and 0.4, respectively. In (b), we can see that the edges of the selected node change their color from black to red when the opacity is above 0.5. We make this color change so that the highlighted node can have a more apparent contrast.

4.4 Quiz Component

Besides the study component (refer to Figure 1), GraphVisual offers a quiz component. An instructor can prepare a quiz using this component to not only test students' general knowledge about graph visualization, but also assess whether students can use the provided interaction functions to retrieve information from the underlying graph data.

As shown in Figure 7, the user interface of the quiz component is similar to that of the study component. The main differences are as follows. For the two display panels, the right one is replaced for the question and answer (Q&A). In the control panel, we remove the function of changing the data set, as only a single data set is associated with a question. Furthermore, we decide to move the data set description section to the top of the control panel for quick attention, as the questions are usually data set specific. GraphVisual supports textual questions with a text box for answering as well as single- or multiple-choice questions associated with a corresponding display panel. The interaction functions in the quiz component are a subset of the functions provided in the study component.



Number of nodes (displayed / total): 62 / 62
Number of edges (displayed / total): 159 / 159

(8 of 22)

What are the pros and cons of the spectral layout when depicting the Dolphins data set compared to the other layouts. Briefly explain.

Briefly state your answer and explain your decision.

Layout: D3-Force-[

Name: Dolphins

Type of graph: Social Network

Description: This graph describes the frequent association amongst dolphins. A node represents a dolphin and edge describes their interaction.

Figure 7: The screenshot of GraphVisual (quiz component). Compared with the study component shown in Figure 1, we mainly replace the right display panel of graph drawing with a panel for Q&A.

4.5 Design Considerations

During the design of GraphVisual, we needed to make several choices following the principles specified in Section 3. To comply with **P1** - Simplicity, we carefully offer the options so that the users can learn effectively without being quickly overwhelmed. However, this design principle conflicts with two other principles in two cases. In the first case, **P1** - Simplicity and **P3** - Interactivity do not agree with each other when it comes to graph layouts. It is imperative to support multiple graph layouts as this is the primary purpose of designing GraphVisual. However, allowing interactive parameter adjustment for every layout could quickly overwhelm the users. We, therefore, opt to allow adjusting the parameters and re-animating only one of the force-directed layouts instead of enabling such interactions for all of them. In this way, the main goal of understanding the differences between force-directed and non-force-directed layout algorithms can still be achieved. In the second case, **P3** - Interactivity and **P4** - Comparison have a conflict when we debate whether or not to include the edge bundling function. Following **P4** - Comparison, one interesting comparison would be visualizing a graph with and without edge bundling. Edge bundling [17] has proved to be a flexible method that can be used in conjunction with existing graph visualization to reduce visual clutter and highlight important patterns that may be hidden otherwise. This function is most effective when it comes to visualizing very large graphs. However, computing edge bundles for a large graph would take a significant amount of time, which not only impacts **P3** - Interactivity but also demands resources beyond the ones a typical web-based program could have. As the drawback and constraint outweigh the potential benefits, we decide not to incorporate the edge bundling function into GraphVisual.

5 User Study

We conducted a user study to evaluate the effectiveness of GraphVisual with two different groups. Students in the first group, *in-class group*, did the study while taking the Data Visualization course and had 13 participants, including undergraduate and graduate students from various fields of study. The 10 students in the second group, *PhD group*, did not take the Data Visualization course and are doctoral computer science and engineering students. The study consisted of two parts, an online survey and a quiz using GraphVisual. The survey was designed to receive detailed feedback about whether GraphVisual improved the students' learning of graph visualization. The quiz consisted of 22 questions querying the students' knowledge about graph visualization and examining their ability to understand different graph layouts and extract information from different data sets. The tool automatically recorded student answers and auto-grades everything except for the open text questions. As these open text questions need to be graded manually, we did not provide immediate feedback to the students.

5.1 General Information

The in-class group performed the study as the students took the Data Visualization course at the University of Notre Dame, while the PhD group received the learning materials via email. The students used their laptops to access GraphVisual via their preferred web browser. The in-class group students were informed that this is a pilot project and that the quiz would only count towards their participation grade of the class, while the PhD group students received monetary compensation ($30 each) for their time. We did this to avoid having the students not paying close attention to their answers. The in-class group students include 7 computer science and engineering (CSE) students (4 undergraduate, 1 master, and 2 PhD students), 2 applied and computational mathematics and statistics (ACMS) students (1 undergraduate student and 1 PhD student), 3 non-degree students (1 visiting student, 1 exchange student, and 1 University employee), and 1 aerospace and mechanical engineering (AME) PhD student. They all knew what graphs were before taking the course, and have learned about graph visualization within the class. Five of the students in the PhD group conduct research related to graphs and networks, while the other five do not. To facilitate the process, we designed a study guide to help the students explore GraphVisual. After an introduction session for the in-class group students, they received the study guide and were asked to complete the survey a few days later. Respectively, the PhD group students received an introductory email including lecture slides, a link to GraphVisual, and the study guide, and they received the survey request a few days later. Both groups were asked to complete the study guide before filling in the survey. A week later, the quiz was conducted in class for the in-class group students. Similarly, the PhD group students took the quiz a week after receiving the introductory email.

5.2 Questions

The questions for the survey and the quiz can be found in the Appendix. The survey consists of 21 rating questions with a five-point Likert scale from “strongly disagree” (1) to “strongly agree” (5), and an open text question for general feedback. These rating questions are split into four categories: *graph layout* (S1-S8), *data set* (S9-S10), *user interface* (S11-S18)), and *general* (S19-S21). Out of the 22 quiz questions, 11 are a single choice, 2 are multiple-choice, and 9 are open text questions. Seventeen questions are accompanied by a graph that is displayed interactively, and the students could change the layout if applicable. The rest of the questions test the students’ general understanding of graph visualization. Questions with a data set usually aim at examining whether the students’ can extract information from the visualization of the displayed graph data.

5.3 General Findings

Overall, we can report similar findings from both the in-class group and PhD group students. The in-class group performed slightly better than the PhD group, which was expected to due to the circumstances of the study. Attending lectures is better than self-learning, and receiving the participation grade by completing this study can be a motivation for students to study the concepts. Both groups were satisfied with GraphVisual, and the overall feedback was positive. In the following sections, we describe the findings of the survey and the quiz results in detail.

5.4 Survey

Using a survey, we aimed to find out how well GraphVisual was received and where its shortcomings lie. Table 2 shows the average rating (higher is better) and the standard deviation for the four question categories. We can see that GraphVisual was positively received among both student groups with average scores of $4.37/5$ and $4.35/5$ for the in-class group and PhD group, respectively. The most significant discrepancy and also the lowest rating is reached in the graph layout category. Upon closer inspection, the low rating in the PhD group is mainly due to the following survey question S6: “*The Spectral layout helped me understand the nature of the displayed data better.*” While the in-class group rated this question as 4.15 on average, the PhD group evaluated it as 2.9 on average. Because spectral graph layouts are not commonly known, the in-class group students, who learned about it in class, could increase their understanding better than the PhD group students who followed a self-learning approach (we did not control for the self-learning other than running the quiz). This shows that GraphVisual can enhance the understanding of previously learned concepts; however, it may not be sufficient to teach these concepts (unless a more detailed tutorial is provided similar to what is taught in class). Besides, the other categories show that both groups liked the selection of data sets and the user interface design, as they agreed that using GraphVisual helps them learn knowledge about graph visualization.

	Graph Layout	Data Set	User Interface	General	All Categories
Avg Rating	4.37	4.16	4.63	4.31	4.37
Std Dev	0.63	0.66	0.59	0.78	0.67
Avg Rating	3.95	4.30	4.70	4.43	4.35
Std Dev	0.86	0.56	0.51	0.54	0.62

Table 2: The average rating and standard deviation per question category and all categories. Top two rows: in-class group. Bottom two rows: PhD group.

5.5 Quiz

We used GraphVisual's quiz component to evaluate how well students can apply their knowledge about graph visualization to answer questions about graph layout and data set specific questions. Table 3 and Table 4 show the average score and standard deviation for each of the 22 questions split by the two study groups. We can see that both groups scored reasonably well overall with an average score of $82.87/100$ and $78.64/100$ for the in-class group and PhD group, respectively. These results underline the difference between receiving a lecture and self-learning, similar to the results of the survey.

The lowest scores by question can be found in Q18 to Q22. As these questions are not accompanied by a graph, one reason could be the theoretical nature of the questions. Q18 asks for an explanation of a long edge in a force-directed drawing. While many students had a rough idea, most of them could not articulate it in a precise way. Another reason for low scores of Q19 and Q20 is that we scored multiple-choice questions as either correct (all choices are ticked correctly) or incorrect (not all choices are ticket correctly). Closer inspection shows that 13 students among both groups did not think that OpenOrd is a force-directed layout. Similar to this, 7 students among both groups thought that Yifan Hu is a non-force-directed layout. Q21 shows that students of the in-class group are better in naming the "attractive force" by its correct term compared to the students of the PhD group. Similar to this, Q22 shows that the theoretical foundation was better laid out for the in-class group students. This implies that, in a self-learning setting, while GraphVisual helps students understand graph visualization, more material should be prepared for them to learn the theoretical background better.

Question	Q1	Q2	Q3	Q4	$Q5$	Q6	Q7	Q8	Q9	Q10	Q11
in-class group											
Avg Score	100.00	100.00	92.31	92.31	100.00	100.00	100.00	84.62	76.92	84.62	100.00
Std Dev	0.00	0.00	27.74	27.74	0.00	0.00	0.00	37.55	43.85	37.55	0.00
PhD group											
Avg Score	100.00	80.00	90.00	90.00	90.00	100.00	100.00	80.00	90.00	60.00	70.00
Std Dev	0.00	42.16	31.62	31.62	31.62	0.00	0.00	42.16	31.62	51.64	48.30

Table 3: The average score and standard deviation per question for Q1 to Q11.

Question	Q12	Q13	Q14	Q15	Q16	$Q17$	Q18	Q19	Q20	Q21	Q22	Q1-Q22
in-class group												
Avg Score	100.00	100.00	84.62	92.31	76.92	84.62	46.15	23.08	53.85	76.92	53.85	82.87
Std Dev	0.00	0.00	37.55	27.74	43.85	37.55	51.89	43.85	51.89	43.85	51.89	10.52
PhD group												
Avg Score	100.00	90.00	90.00	100.00	100.00	90.00	40.00	30.00	70.00	40.00	30.00	78.64
Std Dev	0.00	31.62	31.62	0.00	0.00	31.62	51.64	48.30	48.30	51.64	48.30	11.35

Table 4: The average score and standard deviation per question for Q12 to Q22, and for all questions (Q1-Q22).

The left table in Table 5 shows the results split by the CSE and non-CSE students as well as graduate and undergraduate students of the in-class group. Interestingly, there was no significant difference with either split of the group, suggesting that GraphVisual can help students with a diverse set of backgrounds or prior knowledge understand and reach the main learning objectives of graph visualization. From the right table of Table 5, we can see the average quiz scores split by question type. One split evaluates the difference between questions that include graphs and questions that do not. We can see a big difference in the scores: questions without graphs having a much lower score compared to questions with graphs. This shows that the students become very familiar with exploring graph visualizations using

	CSE	Non-CSE	Grad	Undergrad
Avg Score	82.47	83.33	81.82	83.77
Std Dev	10.96	11.01	11.13	10.77

	Graph	Non-Graph	Text	Choice
Avg Score	92.31	50.77	86.32	80.47
Std Dev	8.60	19.31	17.53	23.63

Table 5: Quiz results for the in-class group. Left: The average score and standard deviation separated into CSE and non-CSE students, and graduate and undergraduate students, respectively. Right: Quiz results separated by questions accompanied by graph or not, and by text and choice questions, respectively.

	Graph Experts	Non-Experts
Avg Score	81.82	75.45
Std Dev	5.57	15.88

	Graph	Non-Graph	Text	Choice
Avg Score	89.41	42.00	78.89	78.46
Std Dev	15.39	25.73	23.15	25.12

Table 6: Quiz results for the PhD group. Left: The average score and standard deviation separated into graph expert and non-expert. Right: Quiz results separated by questions accompanied by graph or not, and by text and choice questions, respectively.

GraphVisual; however, it may not necessarily help them learn the theoretical background better. The other split evaluates whether there is a difference between text-based questions and choice-based questions. While students scored better on text-based questions, the difference is not significant.

Table 6 shows two tables with similar splits for the PhD group. On the left, we see the scores split between graph experts and non-experts. As expected, doctoral students whose research focuses on graph and network data, perform better on average than the remaining ones. However, the gap is not dramatic, and the standard deviation indicates that there is a higher difference in the performance of non-experts. Upon closer inspection, we found that one of the non-expert students scored only 47.83, which explains why the average score for non-experts is lower. Removing this outlier raises the average score of the non-experts to the same level as that of the graph experts. This, again, shows that GraphVisual does not require any prior graph knowledge to be used successfully by self-learners. The right table of Table 6 shows the scores split by question type again. Similar to the in-class group, the questions without a graph seem to trouble the students more. In this case, the impact is even worse, suggesting that theoretical graph knowledge should be probably taught in the form of a lecture to ensure learning effectiveness.

6 Conclusions and Future Work

We have presented GraphVisual, a web-based interactive tool to assist in teaching and learning graph visualization. GraphVisual consists of two components (a study component and a quiz component) and supports various graph layouts and data sets of different sizes and characteristics. We demonstrated the effectiveness of GraphVisual by conducting a formal user study with two different groups (in-class group and PhD group). In the future, we would like to improve GraphVisual as follows. First, so far only a few layouts allow dynamic changes and re-computation, which could be expanded to the other layouts. Second, one participant in our study suggested that GraphVisual could point out and explain differences in the different layout algorithms when comparing them side by side, which could be included as an additional feature for self-learning. Third, GraphVisual could still incorporate edge bundling; however, based on our experience, this would need to be outsourced to a server that runs the computations in the background.

GraphVisual is part of our effort to develop a series of educational tools for teaching and learning data visualization. We have released GraphVisual online `https://sites.nd.edu/chaoli-wang/visvisual`, and the source code can be downloaded from `https://github.com/mimre25/graph-visual`. Besides previously developed FlowVisual [26, 25] and GraphVisual, we will further develop other tools for scientific visualization and information visualization, under the umbrella of VisVisual. Each tool of VisVisual can be used independently, either for classroom teaching and learning or self-learning.

Acknowledgements

This research was supported in part by the U.S. National Science Foundation through grant DUE-1833129. We would like to thank the following students who contributed to the GraphVisual development: Alexander Mascoli, Ahmed Farag, Kekoa Wong, and Kyle Weingartner.

References

[1] The 2015 Gordon Research Conference on Visualization in Science and Education. `https://www.grc.org/visualization-in-science-and-education-conference/2015/`. Accessed: 2019-10-16.

[2] J. Barnes and P. Hut. A hierarchical $O(n \log n)$ force-calculation algorithm. *Nature*, 324(6096):446, 1986.

[3] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An open source software for exploring and manipulating networks. In *Proceedings of International AAAI Conference on Weblogs and Social Media*, 2009.

[4] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*, 36(1):133–159, 2017.

[5] J. Berry. Improving discrete mathematics and algorithms curricula with LINK. *ACM SIGCSE Bulletin*, 29(3):14–20, 1997.

[6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[7] M. Bostock, V. Ogievetsky, and J. Heer. D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[8] Y. Carbonneaux, J.-M. Laborde, and R. M. Madani. CABRI-Graph: A tool for research and teaching in graph theory. In *Proceedings of International Symposium on Graph Drawing*, pages 123–126, 1995.

[9] C. Chen. Top 10 unsolved information visualization problems. *IEEE Computer Graphics and Applications*, 25(4):12–16, 2005.

[10] R. F. Cohen, A. Meacham, and J. Skaff. Teaching graphs to visually impaired students using an active auditory interface. *ACM SIGCSE Bulletin*, 38(1):279–282, 2006.

[11] J. Díaz, J. Petit, and M. Serna. A survey of graph layout problems. *ACM Computing Surveys*, 34(3):313–356, 2002.

[12] T. Dwyer. Scalable, versatile and simple constrained graph layout. *Computer Graphics Forum*, 28(3):991–998, 2009.

[13] J. Ellson, E. Gansner, L. Koutsofios, S. C. North, and G. Woodhull. GraphViz - open source graph drawing tools. In *Proceedings of International Symposium on Graph Drawing*, pages 483–484, 2001.

[14] S. Hachul and M. Jünger. Drawing large graphs with a potential-field-based multilevel algorithm. In *Proceedings of International Symposium on Graph Drawing*, pages 285–295, 2004.

[15] S. Hadlak, H. Schumann, and H.-J. Schulz. A survey of multi-faceted graph visualization. In *Proceedings of Eurographics Conference on Visualization - State of The Art Reports*, pages 1–20, 2015.

[16] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

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

[18] Y. Hu. Efficient, high-quality force-directed graph drawing. *The Mathematica Journal*, 10(1):37–71, 2005.

[19] M. Imre, J. Tao, Y. Wang, Z. Zhao, Z. Feng, and C. Wang. Spectrum-preserving sparsification for visualization of big graphs. *Computers & Graphics*, 2020. Accepted.

[20] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian. ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PloS One*, 9(6):e98679, 2014.

[21] S. Khuri and K. Holzapfel. EVEGA: An educational visualization environment for graph algorithms. *ACM SIGCSE Bulletin*, 33(3):101–104, 2001.

[22] S. Martin, W. M. Brown, R. Klavans, and K. W. Boyack. OpenOrd: An open-source toolbox for large graph layout. In *Proceedings of IS&T/SPIE Conference on Visualization and Data Analysis*, page 786806, 2011.

[23] Y. Tanahashi, N. Leaf, and K.-L. Ma. A study on designing effective introductory materials for information visualization. *Computer Graphics Forum*, 35(7):117–126, 2016.

[24] C. Wang and J. Tao. Graphs in scientific visualization: A survey. *Computer Graphics Forum*, 36(1):263–287, 2017.

[25] M. Wang, J. Tao, J. Ma, Y. Shen, and C. Wang. FlowVisual: A visualization app for teaching and understanding 3D flow field concepts. In *Proceedings of IS&T Conference on Visualization and Data Analysis*, pages 476–1–476–10, 2016.

[26] M. Wang, J. Tao, C. Wang, C.-K. Shene, and S. H. Kim. FlowVisual: Design and evaluation of a visualization tool for teaching 2D flow field concepts. In *Proceedings of American Society for Engineering Education Annual Conference*, pages 23.609.1–23.609.20, 2013.

A Appendix

A.1 Survey Questions

Graph layouts

S1 The number of layouts provided by GraphVisual is sufficient for me to learn different graph layout algorithms.

S2 GraphVisual helped me better understand the differences among graph layout algorithms.

S3 The baseline randomized and circular layouts helped me compare and better understand the advantages of other layouts.

S4 GraphVisual helped me better distinguish force-directed and non force-directed graph layouts.

S5 For the D3-Force-Directed layout, “Attractive Force” and “Repulsive Force” sliders in the interface helped me better understand the roles of these two counterforces in the force-directed layout algorithm.

S6 The Spectral layout helped me understand the nature of the displayed data better.

S7 The special layouts (geographical or bipartite) helped me better understand data-specific layout requirements (e.g., for graph data having geographical information or disjoint node groups).

S8 GraphVisual helped me better analyze which graph layout works better for a given data set.

Data sets

S9 The data sets provided are representative which helped me better learn and differentiate different graph layout algorithms.

S10 The data sets provided in each category (small, medium, and large) are sufficient for me to learn graph visualization.

User interface

S11 The on-boarding feature helps me better get oriented with GraphVisual.

S12 The “about” page provides me with sufficient information to gain an overview of GraphVisual.

S13 The overall interface design of GraphVisual is intuitive and easy to follow.

S14 The default node size and edge thickness are appropriate.

S15 The use of colors in the visualization clearly distinguishes different items.

S16 The visual effects for user interface elements help me better distinguish disabled items or controls.

S17 The interactions are sufficient for serving the basic graph navigation need.

S18 The interactions along with the given tooltips are easy to understand and follow.

General

S19 The questions provided in the study guide help me learn GraphVisual.

S20 GraphVisual provided an enjoyable way to learn about graph visualization.

S21 Using GraphVisual helped me feel confident in my knowledge about graph visualization.

S22 What changes to the items above, or other parts of GraphVisual, would improve your learning of graph visualization?

A.2 Quiz Questions

Text questions are labeled with a †, single-choice questions with a $\star$, and multiple-choice questions with a $\sim$.

Q1 $\star$ Assuming that the edge length in the drawing represents the length of a fiber tract, which two nuclei are the farthest apart?

Q2 † Which of the following two layouts (ForceAtlas2, OpenOrd) is better when depicting the Cat Brain data set? Briefly explain.

Q3 $\star$ How many states have every legal ground for divorce?

Q4 $\star$ Among the states listed below, which state does not have the same legal grounds for divorce as others?

Q5 $\star$ Among the states listed below, which state has fewer legal grounds for divorce than others?

Q6 † What are the pros and cons of the bipartite layout when depicting the Divorce data set? Briefly explain.

Q7 ⋆ Which node is the most distinct one in the spectral layout?

Q8 † What are the pros and cons of the spectral layout when depicting the Dolphins data set compared to the other layouts. Briefly explain.

Q9 ⋆ Among all the employees, the one with the most contact has how many contacts?

Q10 ⋆ Which two nodes have the longest path (along graph edges) in the spectral layout?

Q11 ⋆ What percentage of naval bases have connections with at least 8 other bases?

Q12 † Which of the following two layouts (FM3, spectral) is better when depicting the Naval data set? Briefly explain.

Q13 † Among the force-directed layouts, which is the best in depicting the Netscience data set? Briefly explain.

Q14 † Briefly explain the main similarities and differences between OpenOrd and Yifan Hu layouts when depicting the Netscience data set.

Q15 ⋆ Which airport has the most direct flights to and from other airports?

Q16 ⋆ From Anchorage Intl to Pago Pago Intl, which of the following airport should be used for transferring so that the number of legs is minimized?

Q17 † What are the pros and cons of the geographical layout when depicting the USAir97 data set? Briefly explain.

Q18 † In a force-directed graph drawing, what does a long edge indicate?

Q19 ∼ Which of the following are force-directed layouts?

Q20 ∼ Which of the following are not force-directed layouts?

Q21 † In a force-directed layout, what is the name of the force that counterbalances the repulsive force?

Q22 ⋆ What is the spectrum of a graph called?

Data sets used in quiz questions

Q1	Cat Brain
Q2	Cat Brain
Q3	Divorce
Q4	Divorce
Q5	Divorce
Q6	Divorce
Q7	Dolphins
Q8	Dolphins
Q9	Enron Email
Q10	Enron Email
Q11	Naval
Q12	Naval
Q13	Netscience
Q14	Netscience
Q15	USAir97
Q16	USAir97
Q17	USAir97
Q18	None
Q19	None
Q20	None
Q21	None
Q22	None