The Role of Data and Task in Visualization Design and Recommendation Systems

Aditeya Pandey
pandey.ad@northeastern.edu
Khoury College of Computer Sciences, Northeastern University

Abstract

Data visualization is a graphical representation of information and data, enabling end-users to identify meaningful knowledge from the data and make informed decisions. However the visualization design process is intensive and prone to errors. A common pitfall in visualization design is the incorrect mapping between data and task requirements and visual encoding, or in simple terms, "choice of visualization". The mapping of data and tasks to a suitable visual encoding is challenging for the following reasons: (1) Limited Theoretical Support: Existing visualization theory may offer limited support to identify the analytical goals of the end user, (2) Incomplete or Inconsistent Guidelines: The knowledge for designing visualizations may be incomplete, inconsistent or inaccessible, (3) Lack of Practical Resources: Shortage or lack of resources that can guide visualization practitioners in their task of visualization design. This dissertation aims to mitigate these challenges and support visualization practitioners and researchers in selecting visualization design more effectively.

The overarching research goal of my thesis is: *How can we develop theory, identify visualization best practices, and build applications that enable visualization practitioners to create effective and expressive visualizations?* To answer this question, the proposed dissertation contributes knowledge to the field of visualization in three ways: theory, design guidelines, and recommendation systems. The theory contribution includes identifying shortcomings of existing visualization theory and recommending methods to eliminate the shortcomings. To determine the shortcomings, I will conduct three novel design studies CerebroVis (already complete/published), Strokevis and Segmentrix+Portola (in prep for publication). In these studies, I will develop tree and network visualizations to solve novel domain problems. A posthoc analysis of the studies completed so far have revealed that existing general task abstraction theory lacked specificity to describe tree visualization tasks. To improve the task abstraction specificity for trees, this dissertation contributes a tree-specific extension to the established task abstraction framework Multi-Level Task Typology. This dissertation also contributes visualization design guidelines. In my work, I systematically curate task-based design guidelines through a meta-analysis of empirical results of tree visualization effectiveness from a survey of over 50 papers. In addition to tree visualization, this dissertation also contributes novel design guidelines for data glyphs and timelines. The final contribution of the thesis is focused on making the design guidelines and visualization knowledge easy to access for practitioners and researchers. To do so, I contribute two novel visualization recommendation systems. Based on the theory and data gathered on trees, I will develop a recommendation system to help visualization practitioners navigate the vast tree design space and choose an effective visual encoding based on their data and tasks. In addition to the tree visualization recommendation system, this dissertation also contributes a knowledge-based recommendation system for genomics visualizations. Such systems would make tree and genomics visualization creation accessible to both experts and novices and improve visualization literacy.

Through a series of theoretical and practical contributions, my thesis supports mapping data and task requirements to appropriate tree and genomics visualizations. In this thesis, I also reflect on the broader applications of the contributions, specifically how my work can act as a framework to support the effective and expressive design of information visualization in general.

Contents

1 Introduction

Data visualization enables end-users to analyze data visually. The use of visualization is pervasive and plays a critical role in revealing meaningful insights from the data in different fields. For example, machine learning and artificial intelligence depends on data visualization to understand the learning algorithm's underlying working. In clinical settings, data visualization also enables physicians to visually analyze health indicators like the patients' heart rate. The Nested Model of Visualization Design and Validation [32] and Design Study Methodologies [45, 49] are commonly used visualization design models and frameworks that support practitioners and researchers to create visualizations. These models and frameworks argue that the core of visualization design is mapping data and task requirements of a domain problem to suitable visualization encodings and interaction techniques. For example, an epidemiologist may want to analyze the patterns in the branching of a virus strain and compare how different strains evolve. The epidemiologist is analyzing hierarchical data and the task that they want to accomplish is related to comparing the branches of the tree. Based on the data and the task requirements of the epidemiologist the most appropriate visualization technique is the node-link tree visualization.

The process of mapping data and task requirements is intensive and prone to errors [45]. An error in the mapping can lead to an ineffective choice of visualization [29, 32]. Ineffective visualization design can lead to the spread of misinformation by distorting the way information is perceived by people [50]. Based on my personal experience with designing visualizations for different domain problems [38, 40] and analyzing existing visualization theory [45] I found that process of mapping data and task requirements to visualization encoding can be broken down into four steps. I refer to the four steps combined as the **"Visualization Design Pipeline"** (Fig. 1). Each step in the pipeline is an action that a visualization practitioner has to perform to go from the data and task requirements to a finished visualization product. In Fig. 1, the top row shows the stages of the visualization design pipeline. Data and task abstraction (S1) identifies the data and task requirements of the problem. Practitioners use the data and task requirements to identify an appropriate visualization technique. To determine the visualization technique, practitioners have to identify meaningful design guidelines (S2), and based on the guidelines, choose the appropriate visualization design (S3). The final step in the pipeline is implementing the visualization for the end-user (S4). In Fig. 1, the bottom row shows the output for each step in the task of designing a tree visualization. The output of S1 are abstract data and task specification. Output of S2 represents necessary design guidelines for the identified data and task requirements. Using the guidelines in S3 practitioners choose an appropriate visualization encoding. A polished version of the visualization is implemented for analysis in S4. I describe the steps below and also identify the challenges they can pose for visualization practitioners:

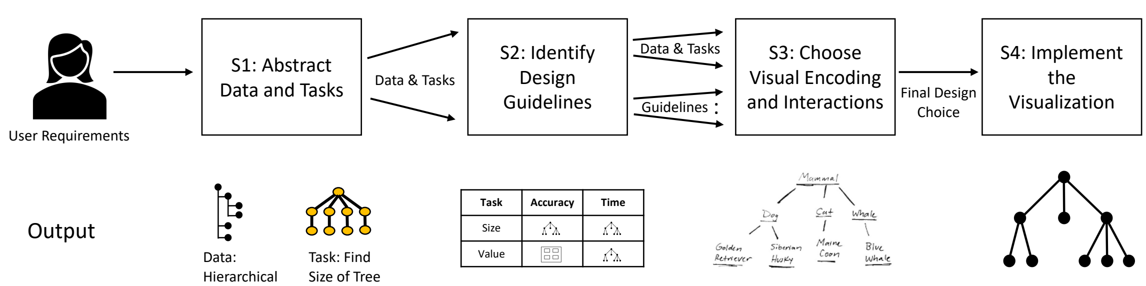


Figure 1: Visualization Design Pipeline: The top row shows a typical visualization design pipeline. The bottom row shows the output for each step in the task of designing a tree visualization.

1. **Abstract Data and Tasks:** The data and task abstraction phase of the visualization design pipeline maps the observed data and domain goals to generalizable abstract data and task specifications using visualization theory. For example, a biologist may be interested in results for tissue samples treated with LL-37 matching up with the ones without the peptide. A visualization designer may identify that results of the experiment are recorded as **numerical** values and translate the task to **comparison** of

values between **two groups**. This transformation of data and tasks from domain-specific to abstract language is essential to enable visualization creators to effectively compare data and tasks across different domains and look for relevant techniques and strategies in different application areas.

Challenges: Visualization researchers have proposed various data and task abstraction approaches (e.g., [2, 6, 27, 33, 46]). Adopting an appropriate abstraction approach is pivotal for visualization design as it impacts the choice of visualization design and interaction idioms. However, selecting a proper abstraction framework requires an extensive comparison of existing literature. A practical solution to the problem of multiple options is to choose a general-purpose framework. For instance, the Multi-Level Task Typology (MLTT) framework [6] and its extended version in the Visualization Analysis and Design textbook [33] is a generic data and task abstraction framework that works well across disciplines and dataset types. However, the general-purpose frameworks sometimes lack the specificity to support task abstractions for specific dataset types such as temporal, spatial-temporal, networks, and trees.

2. **Identify Design Guidelines:** Chen et al. define "a guideline embodies wisdom advising a sound practice in creating a visualization image, designing a visual representation, or developing a visualization system." [10]. Design guidelines can be general like the "overview first and details on demand" mantra by Shneiderman [46], or specific to data type and task such as to the comparison of quantitative data is more effective with position and length visual channels (Clevland & McGill [11]). In this step, a visualization practitioner has to identify the important design guidelines for their visualization problem. This step is manual and currently requires visualization practitioners to scan and analyze information spread across scientific papers, textbooks, and technical reports.

   Challenges: The scientific community has amassed a wealth of empirical knowledge, case studies, tools, and techniques over the past decades. However, most of the knowledge is spread across scientific papers, inaccessible to the general audience of designers, and visualization practitioners [13]. The Visguides project [13], aims to resolve this problem by creating a platform where visualization researchers can share, discuss and critique design guidelines. However, the platform is still in an early phase and only has a small subset of design guidelines. Another major problem for this step is incomplete knowledge. Visualization literature is evolving, and there are many areas and aspects of visualization that have no design guidelines to support visualization practitioners. This opens up opportunities for visualization researchers to fill the gap and build visualization knowledge.

3. **Choose Visual Encoding and Interactions:** Visualization practitioners use the data and task abstraction and the design guidelines to identify appropriate visual idiom and interaction techniques. As discussed previously, the epidemiologist analyzing virus strains' evolution maps the data and task requirements to a node-link tree visualization over other forms of information visualization techniques. Visual encoding and interaction mapping allow practitioners to winnow down from a large visualization design space to find visualization technique/s that match users' goals while respecting the visualization design guidelines.

   Challenges: To select visual encoding and interaction, a visualization practitioner needs to collate and analyze large amount of information about visualization techniques and design guidelines from existing literature scattered in books, empirical studies, and survey reports. This step's manual nature, combined with large design space and unstructured availability of guidelines, can lead to a sub-optimal visualization encoding or an incorrect visual encoding to represent the data. Recently, visualization recommendation tools like Draco [31] and Voyager [55] have tried to reduce the workload from a visualization practitioner by recommending them appropriate visualization encoding based on data and task requirements. However, existing tools do not support visualization techniques like trees and networks or domain specific visualizations like visualization for genomics.

4. **Implement the Visualization:** Visualization practitioners implement the visualization encoding and interactions to a working prototype or a tool to enable users to analyze data and perform the visualization tasks. To implement a visualization design, practitioners can use a spectrum of tools ranging from low-level visualization libraries like d3.js [5] to high-level visual analysis tools like Tableau [51] and Power BI [41]. The low-level visualization libraries are flexible and enable practitioners to create

customized visualization, but they can be hard to learn. Tools like Tableau and Power BI allow practitioners to develop visualization without programming, but they only support a handful of visualization techniques limiting the visualization practitioner's expressivity.

Challenges: Given the broad spectrum of tools and the range of functionalities they offer, it is challenging for visualization practitioners to determine the solution conducive to their visualization design problem. The tools usually have manuals that outline their functionality, but the practitioners may still need to employ a trial and error method to select the right tool. The trial and error method is time-consuming. Organizations with time and resources may afford to invest time in this part of the process. Still, it may be a disadvantage for the larger visualization practitioner audience, who may not have the time or resources to invest in this process.

The challenges in each step of the visualization design pipeline are also potential research opportunities. This dissertation focuses on improving the overall visualization design pipeline by eliminating or reducing the challenges. Through a series of research projects, I demonstrate how the challenges can be solved for tree visualizations. I use tree visualization as a case study because it is beyond the scope of a PhD thesis to solve the challenges for the entire visualization design space. It is a career endeavor. The scope of my thesis is also motivated by a wide range of applications for tree visualizations. Tree visualizations of hierarchical data are common in many fields such as software engineering, machine learning, geography, finance, and biology. The typical applications of tree visualizations in these fields are organization and representation of code-bases in software engineering, explainability of decision-tree models in machine learning, presentation of natural geographical phenomenon like river branching in geography, and exploration of genetic evolution data in biology [28, 33]. Therefore, my thesis contributes information on visualization creation challenges accompanied by resources and tools to help visualization practitioners design widely applicable tree visualization techniques. However, my projects are not limited to tree visualizations. Throughout this thesis, I also present visualization projects that solve the visualization design pipeline challenges for glyphs, timelines, and genomics visualizations. More broadly, my thesis also discusses a framework that can support the identification of visualization design challenges and support creating tools and resources for a broad range of visualization techniques.

Contributions: In this thesis, I will present three novel visualization design studies: CerebroVis, StrokeVis, and Segmentrix+Portola. CerbroVis [38] is a novel visualization technique for representing cerebral arteries, and StrokeVis builds over CerebroVis to detect and diagnose stroke in patients. Segementrix+Portola presents a novel visualization system to detect anomalies in the network traffic of a data center. I applied data visualization theory to develop tree and network visualizations to solve the domain problems in all the design studies. A post hoc analysis of these studies to date revealed that existing Multi-Level Task Typology (MLTT) [6, 33] lacks specificity to abstract tree visualization tasks. To enable effective task abstraction for trees, I also contribute a novel extension of the Multi-Level Task Typology to include more specificity to support tree-specific tasks and a systematic procedure to conduct task abstractions for tree visualizations. With an extensive task abstraction theory for tree visualizations, I curate task-based design guidelines for tree visualizations by surveying published empirical studies. In addition to tree visualization design guidelines, this thesis also contributes task-based design guidelines for data glyphs and timelines. Based on the theory and data gathered on trees, I present a recommendation system to help visualization makers navigate the vast tree design space and choose a useful visual encoding based on their data and, importantly, their tasks. Beyond recommendation for trees, I also contribute a system to recommend genomics visualizations called Genorec. A key aspect of Genorec is its ability to guide an analyst to an existing visualization tool to help them implement the visualization.

In the remainder of this proposal, I will discuss and present related work, research questions, a thesis plan, work accomplished to date, and future work. In Related Work, I will discuss the visualization theories, state-of-the-art in recommendation systems and provide background about tree visualizations as they are an overarching topic in this thesis. In Research Questions, I will discuss the research questions that guide my work. In Thesis Plan, I will discuss how I will answer those questions and my thesis milestones. In the remaining sections, I will discuss the thesis's accomplished work and plans for on-going and future projects.

2 Related Work

2.1 Visualization Theory

Information visualization theory provides theoretical tools like models, frameworks, and guidelines to create the visualization. In this section, I will discuss the visualization design models and frameworks. After that, I will discuss different types of data and task abstraction frameworks. Finally, I will provide an overview of the visualization design guidelines and discuss the challenges and opportunities in this area.

2.1.1 Visualization Design Models and Frameworks

The "Nested Model of Visualization Design and Validation" was developed by Tamara Munzner to guide visualization practitioners with the creation and analysis of visualization systems [33]. The model identifies the key steps involved in visualization design, namely: domain characterization of the data and tasks, an abstraction of domain goals into operations and data types, visual design encoding and interaction techniques, and create algorithms to execute these techniques efficiently. Munzner argues that all the model steps are essential and play an indispensable role in designing effective and expressive information visualization. Sedlmair [45] et al. proposed a "Design Study Methodology Framework" (DSM) where they broke down the Nested Model by Tamara Munzner into detailed step-by-step instructions to guide practitioners on how to create a visualization design. The extended DSM framework consequently divides the visualization design into nine stages: learn, winnow, cast, discover, design, implement, deploy, reflect, and write. For each stage, authors provide real-world examples from their own research experience to help visualization practitioners understand the stages better and apply them to their problems. A pivotal step in both the Nested Model and the DSM framework is mapping the abstract data and task definitions to suitable visual encoding and interactions. In this stage, practitioners make decisions about good and bad matches of visual encoding based on their understanding of visualization design theory. While the Nested Model and DSM framework highlight the importance of the data and task to the encoding mapping stage, they do not provide sufficient knowledge to perform the mapping. The knowledge to perform data and task abstraction is available in data and task abstraction frameworks or taxonomies, and the mapping knowledge is available in the visualization design guidelines. Therefore, we will discuss the common and relevant data and task abstraction frameworks and visualization guidelines in the next two sub-sections.

2.1.2 Data and Task Abstraction Frameworks

Data and task abstraction map tasks and data from the specific domain's vocabulary into a more abstract and generic description in the vocabulary of computer science. More specifically, it is in the vocabulary of information visualization. Visualization research provides many abstraction frameworks [3, 33, 24, 44] and taxonomies [1, 2, 27, 30, 36, 46]. Out of the many abstraction frameworks, the most comprehensive is the "Three-part analysis framework" proposed by Tamara Munzner in the textbook Visualization Analysis and Design [33]. The three-part analysis framework provides information on how to abstract data and visualization problems and classify a visualization in terms of marks, channels, and interactions. The data abstraction part of the framework classifies the visualization dataset in terms of dataset type (tree, network, table, spatial data, etc.), the data types (nodes, links, attributes), and the attribute types (categorical, ordinal, and quantitative). The task abstraction part of the three-part analysis framework was adopted from the Multi-Level Task Typology (MLTT) by Brehmer & Munzner [6]. MLTT helps the user understand why a particular task is carried out and breaks down the task into high-, mid-, and low-level categorization along with the final target of the task. In the framework, each categorization consists of abstract concepts to delineate the various objectives at each stage of the task. For instance, the high-level categorizations analyze whether the visualization is used to *consume (discover, present, and enjoy)* or *produce (annotate, record, and derive)* data. The mid-level actions *(lookup, locate, browse, and explore)* describe the type of *search* carried out based on the target and location knowledge. The low-level actions *(identify, compare, and summarize)* represent the type of *query* performed on the target. Targets can be different kinds of *data (e.g., trends, or outliers)*, *attributes (e.g., extremum)*, and *topology (for network data)*.

Existing task abstraction frameworks and taxonomies are either general-purpose or dataset-specific. General-purpose frameworks and taxonomies such as the MLTT [6], Low-level components of analytic activity [2], work well across all disciplines and data-set types. However, they lack the specificity that a dataset-specific task abstraction framework or taxonomy provides. For example, Task Taxonomy for Graph Visualization [27] is a taxonomy for tasks in the field-specific to graph visualizations. This taxonomy provides a more descriptive identification of visualization goals for network visualization tasks than a generalized framework [38]. Therefore, in this thesis, I propose a method to extend a general-purpose abstraction framework (MLTT) (Sec. 6) to include dataset-specific information for trees.

2.1.3 Visualization Design Guidelines

In information visualization literature, guidelines are a set of rules or best practices that guide visualization practitioners to create visualizations. For example, in the classic study by Clevland and McGill [11], authors found out that the representation of quantitative values with position or size channel was more accurate compared to an area or angle channel. Visualization guidelines are scattered in research papers, technical reports, and surveys. Due to scattered availability of resources, visualization practitioner have a hard time to identify and find meaningful guidelines for their visualization problems [13]. The scattered guidelines also make it hard for researchers to identify the data types and tasks that have already been evaluated and the where are the opportunities for novel research [37]. Recently, there have been efforts from the visualization community to build practical resources to share and discuss visualization guidelines. Visguides [13] is a platform that enables visualization researchers to share and critique design guidelines on an open and accessible web platform. Besides Visguides, there are websites like From Data-to-Viz [16] and FlowingData [15] that educate visualization practitioners on the best visualization design practices and help them create visualization by pointing them to the right resources. Although the visualization community is making progress on communication and dissemination of design guidelines, it remains an open question on how do we systematically curate and communicate design guidelines. Therefore, there is vast research potential in this area of information visualization research. My work contributes novel design guidelines for data glyphs, timelines and trees (Sec. 7).

2.2 Visualization Recommendation Systems

Visualization recommendation systems assist practitioners to identify the most compelling visualization technique from a relatively large visualization design space [19, 31, 54, 55]. Due to the growth of design space and increased involvement of people from diverse backgrounds in visualization design, visualization recommendation tools are more important now than ever [53].

Visualization recommendation systems must take into account four considerations [22, 53]: (1) *Data Characteristics* deals with the identification of visual encoding corresponding to data type and attributes. Mackinlay [29] identified the correlation between data attributes and visual encoding. Polaris [48], the research prototype of Tableau software, further adopted the data-based recommendation system by Mackinlay to develop the "Show Me" feature. Voyager [55] is another recommendation system that automatically suggests meaningful visualizations based on the statistical characteristics of the underlying data. (2) *Task Oriented* recommendations factor in users' intentions behind visualizing data as the main criteria for recommending visualization. The current task-based recommendation system support domain-independent low-level analytical tasks like compare, summarize, distribution [42] and domain level tasks [23]. (3) *Domain Knowledge* imposes further restrictions on the results of the recommendation system as the domain expert may prefer a visualization that is more familiar or widely accepted within their domain. Specialized organizations like NASA have developed domain-based recommendations to assist in-situ visual analysis of spacecraft data [25]. (4) *User Preference* relates to factoring end users' preference in the recommendation system output. For instance, in an aesthetic evaluation of tree visualization techniques, researchers found that the sunburst chart was most preferred by the users [9]. Draco [31] has a method to factor user preference in the form of user-defined constraints.

Existing recommendation systems use the recommendation considerations that suits their objective. For instance, tools like Tableau and Voyager [55] only use the *Data* characteristics because they want to recommend a starting point for the visualization practitioner or analyst. Tools like SeedDB [52] and VizML [19] use both *Data* and *Task* characteristics for recommendation but their goal is completely automated visualization recommendation. In order to achieve complete automation, SeedDB and VizML compromise on the visualization techniques and the tasks they support. The tools currently support basic one- or two-dimensional visualization techniques and low-level analytical tasks. Draco [31] is flexible and uses data, task, and user preference axes for the recommendation. However, even Draco is not capable of handling domain-specific problems or recommending specialized visualization encoding like tree visualizations.

2.3 Tree Visualizations

A *tree* is defined as a collection of nodes and links. A *node* is a data structure that can have an identifier (id) and a value. A *link* in a tree is a data structure that connects two *unique* nodes. Similar to the node, a link can also have values associated with them. A key aspect that differentiates trees from graphs or networks is the "hierarchical" relationship that exists within the nodes. Hierarchical relations categorize nodes in a tree dataset as "above" or "parent", "below" or "child", and "at the same level" or "sibling". A *tree visualization* is a graphical representation of a tree dataset.Tree visualizations of hierarchical data, common across many fields of study, are used for critical tasks ranging from the exploration of genetic data of species evolution in biology to the visual analysis of network activity in cybersecurity. As a result, a variety of tree designs are available at the disposal of designers. `Treevis.net` [43] catalogs over 300 techniques and categorizes them on geometric dimensionality ("2D", "3D","hybrid"), visual representation of hierarchy ("implicit", "explicit","hybrid") and node alignment ("axis-parallel", "radial", "free").

Data and tasks play an important role in choosing a tree visualization layout. For instance, if the practitioner wants to display the financial stock market and identify outlier stocks, they will commonly use a Treemap [21], as the market cap data can be aggregated to represent the value of a sector, and treemap representation facilitates the identification of extreme values. The tree visualization tasks are designed to acquire information about the structural and data attributes of a tree. The structural attributes provide information about the "topology" of a tree, and the data attributes provide information about the data associated with the nodes and links of the tree.

Despite the pervasiveness of tree visualizations and the importance of tree visualization tasks for effective visualization encoding, there is little formal theory in the field of data visualization to support the effective characterization of the vast design space as well as the creation of these visualizations based on the data structure and importantly the tasks of the user. While there are many general task taxonomies and ideas of how visualizations are created to support a domain task, these frameworks lack the specificity to support trees. This lack of theory makes it difficult for visualization creators to fully characterize the task space that a tree visualization could support and to design and evaluate novel tree visualizations effectively. Therefore, in my thesis I present a survey of tree visualization tasks, and create a novel task abstraction framework for tree visualization tasks (Sec. 6). I use the task theory to develop guidelines for design of tree visualizations (Sec. 7) and use the guidelines for development of a recommendation system (Sec. 7).

3 Research Questions

As described in the introduction, a core phase of visualization design is mapping data and task requirements of a visualization problem to suitable visual encodings and interaction techniques. The mapping of data and task requirements is an intensive process and requires a visualization practitioner to make a series of decisions to create the final visualization. In Fig. 1 (visualization design pipeline), we present four intermediate steps between the data and task requirements and the final visualization design. Each stage of the pipeline is critical. However, these steps are also prone to challenges like insufficient theoretical support for design, lack of clear design guidelines, and practical tools to assist users in visualization design. An error at any stage of the pipeline can propagate downstream, affecting the final visualization choice. My thesis's primary motivation is the identification of challenges that exist in the pipeline and the creation of theory, resources, and tools that can reduce the errors at the stages.

To identify with the limitations and propose suitable solutions, my dissertation addresses the following overarching question:

"How can we develop theory, identify visualization best practices, and build applications that enable visualization practitioners to create effective and expressive visualizations?"

More specifically, this overarching research question calls for visualization research contributions that can solve the challenges at each step of the visualization design pipeline (Fig. 1). To solve the overarching research questions, I divide the problem into four more specific research questions, where each question maps to a stage in the pipeline:

1. *RQ1: What are the shortcomings of existing visualization theory that can inhibit effective and expressive visualization design?*
   To answer this question, I present three visualization design studies. In these studies, I present how we can apply existing visualization theory to solve novel visualization problems. These studies primarily contribute a visualization tool that allows users to solve critical analytical tasks more effectively in their domain. Further, a posthoc analysis of these studies identifies the shortcomings associated with the existing visualization theory.

2. *RQ2: How can we fix the shortcomings of existing visualization theory while maintaining its advantages?*
   To answer this question, I present a methodology to extend existing visualization theory and resolve their shortcomings. For this research question, we focus on the extension of the task abstraction theory for tree visualizations. In this thesis, I focus on tree visualizations because they are widely applied in many application areas, such as biology, computer science, and geography, and were common across the design studies that we discussed in RQ1. More specifically, we found that the existing task abstraction theory for tree visualizations lacked specificity. Therefore, through this contribution, I enhance the specificity of task abstraction for tree visualization in visualization theory.

3. *RQ3: How can we generate and collect visualization design guidelines and ensure that the guidelines comprehensively map the task and data configurations in visualization theory to appropriate design encodings?*
   To answer this question, I present two empirical studies that generate novel visualization design guidelines for data glyphs and timelines. I also present a methodology to curate design guidelines for visualization techniques with previously published evalution results. For tree visualizaions, I collect visualization design guidelines from a survey of published studies. The tree visualization design guidelines are build over the novel task abstraction theory discussed in RQ2. Through these projects, I contribute empirical knowledge to visualization literature.

4. *RQ4: How can we create tools and systems to help visualization practitioners and researchers access the design guidelines and improve visualization literacy?*
   To answer this question, I present two visualization recommendation systems. The recommendation systems enable practitioners and researchers to choose effective visualization encoding or design based on the data and task requirements. Through these systems we contribute a method to communicate visualization design guidelines that can significantly reduce the human-centered shortcomings of the visualization design pipeline.

By answering these questions, this dissertation will contribute knowledge to the visualization literature on how to make the visualization design process robust which leads to more effective and expressive information visualization design. This work pays special attention to tree visualization because of its pervasive nature in the data analysis and visualization field. However, this thesis's findings can potentially lead to a robust visualization design pipeline for encodings beyond tree visualizations.

4 Thesis Plan

In this section, I present an outline of my thesis chapters and an anticipated timeline to complete the research and write the chapters.

4.1 Thesis Chapters

1. **Introduction:** This chapter will provide an overarching motivation for the thesis research questions and summarize its contributions.

2. **Related Work:** The related work chapter will provide the necessary background of the visualization design theory, describe the existing research in the space of visualization design guidelines curation, and discuss state-of-the-art visualization recommendation systems. This chapter will also explain the limitations of the current research and summarize how this thesis's contributions extend the community's knowledge in these areas.

3. **Novel Visualization Design Studies:** This chapter will present two novel data visualization design studies. The projects solve real-world visualization in domains of medical diagnosis and cybersecurity. I will describe the projects as sub-chapters in this thesis, as outlined below. The projects are used as case studies to demonstrate the advantage of using the Nested Model for Visualization Design and Validation (Munzner 2019) for visualization design studies and identify the shortcomings associated with the theory. This chapter will also discuss the practical challenge of choosing appropriate visualization design idioms due to the way design guidelines are curated and communicated in the visualization literature.

    1 **CerebroVis:** This chapter will present a novel abstract representation of cerebral arteries that is more accurate than the traditional 3D representation in the task of detecting cerebrovascular abnormalities. This chapter will also present a novel framing and definition of the cerebral artery system in terms of network and tree theory and characterize neuroradiologist domain goals as abstract visualization, tree comparison and network analysis tasks.

    2 **StrokeVis:** This chapter will present a novel system to diagnose stroke in patients. This novel representation is built over CerebroVis.

    3 **Segmentrix + Portola:** This chapter will present a novel visualization tool that allows cybersecurity analysts to analyze the hierarchical organization of resources and network connections within these resources at data centers. The visualization design focuses on revealing the anomalous network connection that may lead to network attacks. This chapter will also discuss the role of visualization theory that enabled the translation of cybersecurity analysts' goals to abstract visualization tasks.

4. **Extended Task Abstraction Framework:** This chapter will describes the importance of task abstraction for designing and evaluating visualizations. Task abstraction allows visualization creators to abstract the domain-specific task requirements to abstract visualization specific goals. In a survey and meta-analysis of tree visualization tasks, I found that the existing task abstraction framework for trees only offers limited specificity to describe tree visualization tasks abstractly. Therefore, in this chapter, I describe a task abstraction framework for tree visualization tasks. To supplement the task abstraction, I also contribute a tree visualization task dataset. The task dataset consists of over 200 tasks. All tasks in the dataset were abstracted with the novel abstraction framework and analyzed to better understand the state of tree visualizations. These abstracted tasks can benefit visualization researchers and practitioners as they design evaluation studies or compare their analytical tasks with ones previously studied in the literature to make informed decisions about their design.

5. **Guidelines for Visualization Design:** In this chapter, I will present two methods for building design guidelines. First, I create design guidelines for encodings that do not have existing empirical studies. In this thesis, I present two novel empirical studies that allowed me to build design guidelines for visualizing multi-dimensional data as glyphs and presenting temporal event data as timelines. For visualization techniques with evaluations, such as tree visualization, we curate guidelines by surveying results from published studies and conducting a meta-analysis to identify the general trends and patterns in the results. In the sub chapters of the thesis, I will describe the individual projects.

    1 **Evaluating the Effect of Data Glyphs on Probabilistic Categorization Task:** In this chapter, I will present an empirical study that measures the effect of representing multidimensional data as glyphs for a probabilistic categorization task. This study contributes guidelines for the effective use of glyph designs.

    2 **Evaluating the Effect of Timeline Shape on Visualization Task Performance:** In this chapter, I will present an empirical study that measures the effect of different timeline shapes,

such as linear, circular, and spiral, for representing temporal events data. This study contributes guidelines for the effective use of timeline shapes.

3 **Collecting and Curating Task Based Design Guidelines for Tree Visualizations:** In this chapter, I will present a methodology to curate design guidelines for tree visualizations from previously published empirical studies. I will also discuss the task-based challenges associated with design guidelines curation.

6. **Visualization Recommendation Systems:** This chapter will describe two novel visualization recommendation systems. I will discuss the recommendation systems in detail as sub-chapters of this thesis. I will conclude this chapter with a discussion on common design patterns that emerged from the design of recommendation systems for two very different problems.

   1 **Genorec:** Genorec recommends visualization to biologists and data analysts working with genomics data. In this chapter, I will present the methodology to design and develop a knowledge-based recommendation system for genomics.

   2 **Treevis Recommendation System:** Treevis recommendation system suggests appropriate tree visualization encoding to visualization practitioners or researchers. In this chapter, I will present in detail the methodology to develop a task based recommendation system for tree visualizations.

7. **Discussion:** In this chapter, I will reflect on the challenges faced in solving the research questions and the lessons learned in the process. In the discussion chapter, I will also discuss how we take the lessons learned from this thesis and apply it more broadly in extending visualization theory, curating design guidelines, and building visualization recommendation systems.

8. **Conclusion:** This chapter will reiterate the thesis's primary contributions and argue that I have made significant contributions to the topics of information visualization design and recommendation. I will also urge the community to build on these contributions and make the process of designing and developing visualizations accessible and effective.

4.2 Thesis Completion Timeline

I anticipate to finish my thesis in Spring 2022. Fig. 2 shows projects and the time-frame to complete the projects. Each project's publication status is available in the column "Status" of Fig. 2. For unpublished work, I present the estimated time-frame to complete the research and the tentative publication venue.

Milestone	Status as of 02/01	Spring 2021	Summer 1 2021	Summer 2 2021	Fall 2021	Spring 2022
Chapter 1 (Introduction)		**Write** the chapter				
Chapter 2 (Related Work)		**Write** the chapter				
Chapter 3 (Visualization Design Studies)				**Write** the chapter		
Cerebrovis	**Paper** IEEE Vis 2019, TVCG					
Strokevis			**Research**			**Submit** to IEEE Vis 2022
Segmentrix + Portola	**Poster** IEEE Vizsec 2019 Best Poster		**Submit** to IEEE Vis 2021 Short Paper			
Chapter 4 (Task Abstraction Theory)				**Write** the chapter		
Tree Visualization Task Survey	**Paper** Conditional Accept with Minor Revision TVCG					
Chapter 5 (Design Guidelines)					**Write** the chapter	
Glyph Evaluation	**Poster** IEEE Vis 2019					
*Timeline Shape Evaluation**	**Paper** CHI 2020					
Design Guidelines for Tree Vis		**Research**			**Submit** to CHI 2022	
Challenges for Curating Guidelines	**Paper** BELIV Workshop 2020					
Chapter 6 (Visualization Recommendation Systems)						**Write** the chapter
Genorec	**Poster** IEEE Vis Best Poster 2020	**Submit** to IEEE Vis 2021				
Treevis Recommendation				**Research**		**Submit** to IEEE Vis 2022
Chapter 7 (Discussion)						**Write** the chapter
Chapter 8 (Conclusion)						**Write** the chapter

Figure 2: Thesis Milestones and Timeline. In the Timeline Shape Evaluation(*) paper, I was the second author, thus I will discuss the parts relevant to my thesis.

5 Novel Visualization Design Studies

In this section, I summarize the design study projects of my thesis. These studies helped in identifying the challenges of visualization design pipeline. For each project, I provide an overview of the main contribution, the results and current publication status.

5.1 CerebroVis and StrokeVis

Summary: Arteries in the human brain form a network of blood flow, and a blockage or leakage in this network can lead to life-threatening cerebrovascular conditions such as a stroke or aneurysm. Conventional

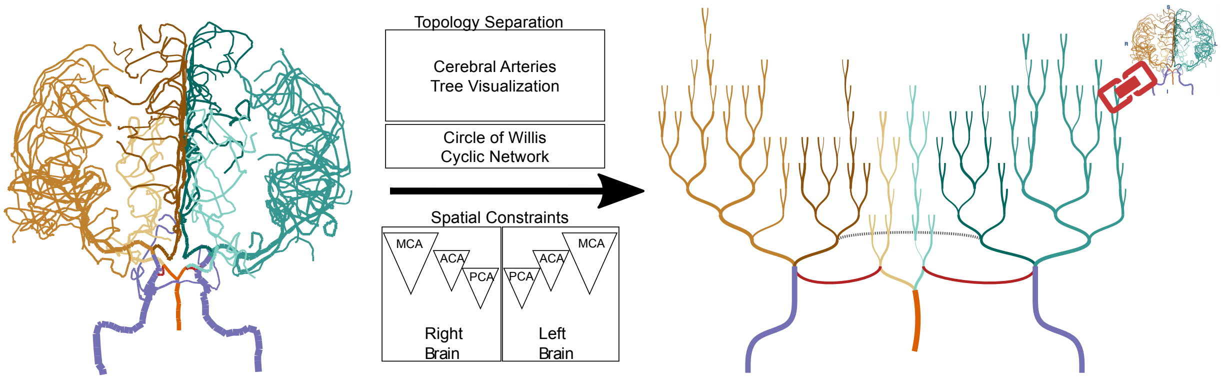


Figure 3: CerebroVis is a novel network visualization for cerebral arteries. CerebroVis uses an abstract topology-preserving visual design which is put in spatial context by enforcing constraints on the network layout. Here we show the conversion of an almost symmetrical healthy human brain cerebral artery network from a 2D isosurface visualization (left) to CerebroVis (right)

diagnostics rely on an expert neuroradiologist identifying vascular abnormalities through examination of medical images (e.g., CTA, MRA). This data is commonly rendered in 3D to assist the doctor with identification of the abnormalities. However, prior research indicates that existing representations of the 3D cerebral arteries—e.g., isosurface, volume rendering, and Maximum Intensity Projection (MIPS)—introduce visual artifacts and task performance challenges such as overplotting/occlusion [14], false impression of geometry [14], and excessive artery bends. In this design study, we present **CerebroVis** a novel 2D visualization of the cerebral artery system with spatial context to assist doctors in the identification of cerebrovascular abnormalities. The abstract visualization enables increased domain task performance over 3D geometry representations, while including spatial context helps preserve the user's mental map of the underlying geometry.

Evaluation and Results: We evaluate our new layout and the accompanying CerebroVis prototype in two ways: (1) assessing the robustness of the technique by examining 61 healthy brain scans and (2) a mixed methods study with three neuroradiologists which included semi-structured interviews and a controlled experiment simulating intracranial stenosis diagnosis. We found that our layout and implementation correctly visualizes all 61 brain scans, that neuroradiologists were more accurate at identifying stenosis with CerebroVis vs. a 3D visualization (absolute risk difference 13%), and that neuroradiologists thought CerebroVis was easy to understand and a useful addition to their diagnosis toolbox.

Status: CerebroVis design study paper was accepted at IEEE Vis 2019 and published in TVCG journal in 2020 [38]. The follow-up work StrokeVis is an upcoming project. StrokeVis will build over CerebroVis layout and focus on detection of cerebral stroke.I plan to conduct the research for StrokeVis in Summer of 2021 and submit it to Vis 2022.

5.2 Segmentrix+Portola

Summary: Micro-Segmentation enables organizations to logically divide the datacenter into distinct security segments down to the individual workload level, and then define security controls for each unique segment. Tree and Network visualizations play a critical role in the development and maintenance of segmentation. In an unsegmented network, a network visualization of workload communication can help domain users assess dependencies and create segmentation policies. Whereas, in segmented networks, the visualization of traffic between individual workloads and segmented groups can be essential for monitoring security compliance. To assist cybersecurity analysts we developed Segmentrix+Portola. Segmentrix is an adjacency matrix-based tool for developing and monitoring micro-segmentation strategies. This representation is scalable, readable, and provides visibility into the entire datacenter network of large organizations. Portola is a radial layered tree visualization, it represents the organization of workloads in a datacenter and assists analysts to visualize traffic through different ports of the network.

Figure 4: Segmentrix+Portola

Evaluation and Results: Segmentrix+Portola was developed over three months at Illumio Inc., with close domain collaboration. Through expert interviews, we found that the visualization system helped find anomalies in network connection and understand the organization of workloads in a datacenter. Segmentrix and Portola were evaluated independently. In the future, we will assess the tools combined as one system.

Status: Segmentrix was accepted as a Poster at Vizsec 2019. It also received the Best Poster Award for Vizsec 2019. I plan to submit a short paper to Vis 2021 which will include both Segmentrix and Portola.

6 Extended Task Abstraction Framework

From the design studies discussed in Sec. 5, I learned that the existing task abstraction theory lacks specificity to abstract tree visualization tasks. For instance, in Cerebrovis, a key task is to analyze symmetry or balance between the left and right brain arteries. However, the existing task abstraction taxonomies do not have the capability to describe the abstraction of a tree balance task. The existing task abstraction frameworks limit visualization researchers and practitioners' ability to define a domain-centric tree visualization task as a well-specified abstract tree visualization task and discover appropriate tree visualization encodings. This section presents a novel extension of the Multi-LevelTask Typology (MLTT) to accurately abstract and analyze tree visualization tasks.

Summary: In the field of information visualization, the concept of "tasks" is an essential component of theories and methodologies for how a visualization researcher or a practitioner understands what tasks a user needs to perform and how to approach the creation of a new design. In this project, I focus on the collection of tasks for tree visualizations, a common visual encoding in many domains ranging from biology to computer science to geography. In spite of their commonality, no prior efforts exist to collect and abstractly define tree visualization tasks. I present a literature review of tree visualization papers and generate a curated dataset of over 200 tasks. To enable effective task abstraction for trees, I also contribute a novel extension of the Multi-Level Task Typology to include more specificity to support tree-specific tasks as well as a systematic procedure to conduct task abstractions for tree visualizations. All tasks in the dataset were abstracted with the novel typology extension and analyzed to gain a better understanding of the state of tree visualizations. These abstracted tasks can benefit visualization researchers and practitioners as they design evaluation studies or compare their analytical tasks with ones previously studied in the literature to make informed decisions about their design. I also reflect on our novel methodology and advocate more broadly for the creation of task-based knowledge repositories for different types of visualizations.

Results: Fig. 5 presents the novel tree-specific extension to the MLTT. The tasks of tree visualization tasks broken down by *Actions* and *Targets*. The Actions use the Multi-Level Task Typology terminology to identify the types of actions users can perform in tree visualization tasks. The Targets include a novel *Nested-extension* of the existing MLTT target characterization that adds specificity for tree visualizations.

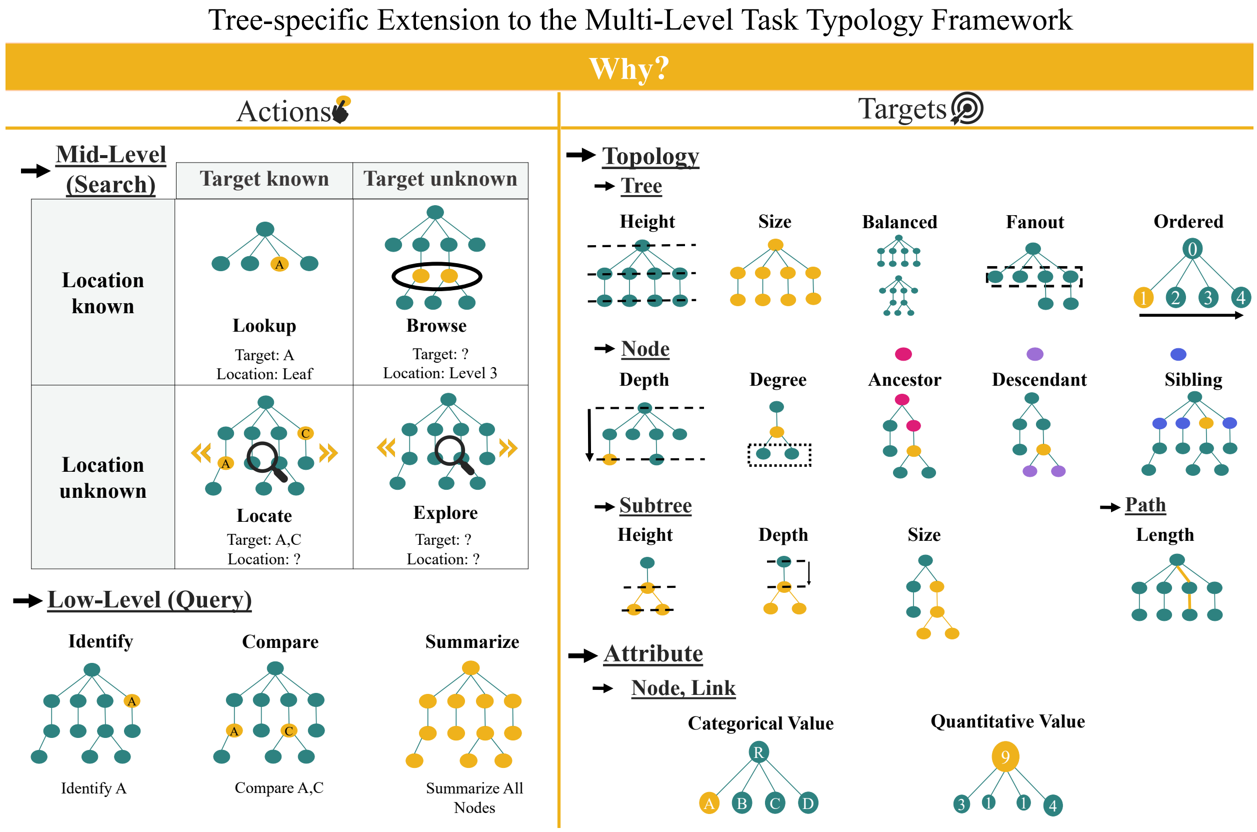


Figure 5: Task Abstraction Framework for Trees

To demonstrate the advantage of the extended task abstraction framework, let us consider the following examples: A financial analyst may want to analyze performance of stocks in different sectors of the market, particularly in the aftermath of a major global crisis to identify outliers that survived recessions and performed well despite an economic slowdown. In another instance, an epidemiologist may want to analyze the patterns in the branching of a virus strain and compare how different strains evolve over time. Applying the MLTT framework for the abstraction for the examples leads to the same abstraction result: the goal of the task is to perform “lookup” on the “topology”. However, in practice the two tasks are fundamentally different. The differences become apparent when we use the extended MLTT framework (Figr. 5), and the abstraction reveals that the financial analyst is interested in “looking up” the “ancestor” nodes of outlier stocks in the stock market tree, and the epidemiologist wants to “compare” properties like “height” and “fanout” of different branches in the tree. Therefore, the novel task abstraction framework enables tree visualization practitioners to be more specific with task abstractions, leading to effective design choices.

Status: The paper is conditionally accepted to TVCG journal and is available on our survey website: `https://intervis-projects.ccs.neu.edu/Tree-Visualization-Survey/`.

7 Guidelines for Visualization Design

This section presents design guidelines for three visualization techniques: data glyphs, timelines, and trees. In my research, I found that there are two common methods to build design guidelines. The first method involves conducting an empirical study, where the researcher evaluates a visualization technique with a set of tasks to identify their effectiveness [12, 26]. In the second method, the researcher surveys previously published studies, and through a meta-analysis of the results, curates guidelines for visualization techniques [17]. I summarize two novel empirical studies for developing guidelines for data glyphs and timelines. There are many tree visualization evaluation studies in information visualization research (for e.g. [8, 26, 47]). However, the task-based design guidelines for tree visualizations are not yet curated [37]. Therefore, for tree visualizations, I describe my plan for curating task-based design guidelines.

7.1 Data Glyphs

Summary: Categorization involves classifying objects based on their features. For example, sorting laundry before putting it in the wash by color and material. Glyphs are visualizations well suited to represent categorization data because they are used for multidimensional data in which dimensions are encoded to marks in the visual or pictorial representation. However, there has been no study or systematic evaluation of how to best encode probabilistic categorization data with glyphs to date. In this study, I evaluate whether the visual representation of the data affects categorization accuracy? And if so, how should the data be visually represented to maximize categorization accuracy? Previous research work has demonstrated that the inclusion of representations of people and human faces results in a significant improvement in memorability for natural images [20] and data visualizations [4]. Research in visualization has also shown that pictorial data encodings are recalled more accurately than simple bar charts [18] when working memory is under heavy load. Consequently, we hypothesized that a memorable glyph representation would result in a higher categorization accuracy for the second question. To test the hypothesis, we evaluated the effectiveness of anthropomorphic (human-like) glyphs as compared to abstract glyphs. In order to evaluate the effect of glyph representation on categorization accuracy, we conducted a within-subject study with 480 participants on Amazon's Mechanical Turk. Each participant completed a probabilistic categorization task with two of four different glyph designs each of which encode 3 probabilistic features. Two of the glyphs were of abstract design and two of the glyphs were human-like so that we could observe whether there was a positive benefit to the more memorable anthropomorphic glyphs, see Fig. 6 (Glyphs Evaluated).

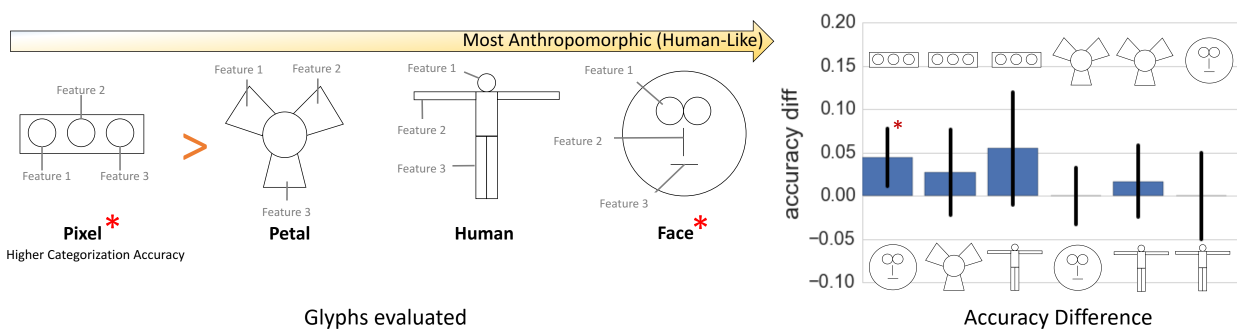


Figure 6: The left figure shows the glyphs evaluated in our study. The pixel and petal glyphs are abstract glyph designs and the human and face are anthropomorphic glyph designs. The figure on the right is a summary of the average differences in accuracy between each pair of glyphs evaluated in the study. On the y-axis, positive ratios denote that glyphs on the top of the chart had greater accuracy, and visa versa for negative. For each glyph comparison, the 99.9% C.I. is plotted, and asterisks (*) denote Bonferroni-corrected significance in accuracy of one stimulus over other.

Results: Contrary to our hypothesis, I found participants were significantly more accurate with abstract than anthropomorphic glyphs, see Fig. 6 (Accuracy Difference). The Pixel glyph visual encoding generated the most accurate categorization performance and lead to statistically significantly higher accuracy than the Face glyph. In addition, participants felt less confident with anthropomorphic glyphs in comparison to the abstract glyphs when performing the categorization task.

Status: The research is complete and the paper is written. The work received initial validation from the community as it was accepted as a poster at IEEE Vis 2019. I plan to submit the paper to an annual visualization workshop VisxVision.

7.2 Timelines

Summary: A timeline is a visual representation of a series of events in time. Timelines have become prevalent in our daily lives as the de facto representation to show financial trends, weather details, and meeting schedules. Timelines are most commonly drawn linearly [7], where the events are organized along a straight line. In practice, however, we can find abundant examples of timelines where events are arranged in non-linear shapes like circles, spirals, grids, and other arbitrary arrangements [7]. The visualization literature provides sufficient evidence that the layout and orientation of visualizations affect user's analytical task performance [11]. However, existing work in timeline visualization evaluation has not measured the impact of timeline shape alone on user task performance for general temporal event sequence data. In this paper, I present the first study which evaluates the readability of timeline shape alone on user task performance for general temporal event sequence data. In a crowd-sourced experiment, I compare 4 timeline shapes — horizontal line, vertical line, circle, and spiral — using 3 types of temporal data — recurrent, non-recurrent, and mixed. Our study is carefully designed to evaluate timeline shapes using common everyday tasks with familiar-looking datasets. E.g., find the date associated with an historical event on a timeline or lookup your daily schedule to find what are you supposed to do tonight at 8pm. In a within-subjects study design, I measured time to complete a visualization task and the task accuracy across the 4 timeline shapes.

Results: There was evidence that task completion time is dependent on the choice of the timeline shape. Specifically, linear shapes were on an average faster to read. But, no evidence supports that users' accuracy is affected by the shape of timeline. Additionally, there was a strong preference for linear timeline among the participants.

Status: This evaluation study was accepted at CHI 2020 [12].

7.3 Tree Visualizations

Motivation and Preliminary Research: Task-based guidelines are critical for the design of a visualization. For instance, if the user's task is to visualize the tree topology, then the preferred visualization design choice should be a node-link tree visualization as they show the topology of the tree explicitly. In information visualization theory, there are many task-based tree visualization evaluation papers (for e.g. [8, 26, 47]). However, these studies have never been collated and analyzed to create consistent task-based design guidelines for tree visualizations. Lack of task-based design guidelines inhibits visualization practitioners from navigating through the design space of tree visualizations and identifying the most effective visual encodings based on their tasks. Therefore, I plan to conduct a meta-analysis of the published empirical studies for tree visualizations and use them to generate task-based design guidelines. In my previous work, as discussed in Sec. 6, I surveyed empirical studies and also identified an exhaustive list of tree visualization tasks. Both these contributions are critical for curating the design guidelines, as explained in the research plan.

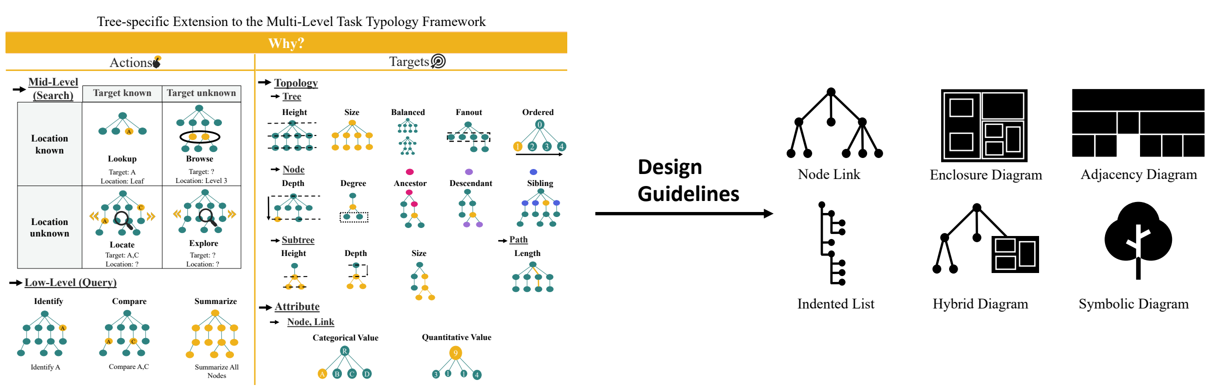


Figure 7: Task-based design guidelines help visualization practitioners to choose the most appropriate visualization encoding based on tasks of the user.

Research Plan: Task-based design guidelines assist visualization practitioners in choosing an appropriate visualization given the task requirements of the user (see Fig. 7). To build the guidelines, I will collect the tasks, the visualization encoding, and results from tree visualization empirical studies. After collecting the tasks, I will abstract the tasks using the Tree-specific task abstraction framework developed (see Fig. 5). The abstraction will enable the comparison of tasks collected from different studies. After the abstraction, I will identify the most appropriate visualization encoding for each task type based on the results of the empirical studies. To find the most appropriate visualizations, I will rank visualization techniques for each task based on consensus. The consensus in ranking will be reached if most empirical results favor a particular visualization encoding for a particular task. For tasks that do not have published results or ambiguous results, the ranking will be built through expert ranking. The expert ranking method was previously used by Nobre et al. [34] to develop a recommendation wizard for Multi-variate network visualizations. In this method, a group of experts ranks visualization encodings based on their task suitability. The experts use their research experience to perform the ranking. The expert team will include expert tree visualization researchers: Michelle Borkin and John Alexis Guerra Gomez.

Expected Outcomes: This work will yield design guidelines that will help researchers, designers, and students to create tree visualizations. The guidelines will also help in the development of a tree visualization recommendation system. The guidelines will be the core of the recommendation model, which will suggest tree visualization encoding to the users. I will discuss the recommendation system in Sec. 8.

8 Visualization Recommendation Systems

In this section, I present two visualization recommendation projects. The first project is a recommendation system for genomics visualizations. The genomics recommendation project is in the last stage of development and validation. I have already shared the initial results with the visualization community by publishing a poster [39]. Therefore, I will discuss the completed research and provide information about how I plan to conduct the remaining work for the genomics visualization recommendation system. In this section, I will also present a recommendation system for tree visualizations. The tree visualization recommendation system project builds over my research in Sec. 6 (Task-Abstraction Theory) and Sec. 7 (Tree Visualization design guidelines). The task abstraction theory helps in guidelines research, and the guidelines will ultimately form the knowledge for the recommendation system. Since I have not started my work on the tree visualization recommendation project, I describe the research plan, expected outcome, and validation plan in detail.

8.1 Recommendation System for Genomics Visualization

Summary: Visualization tools and techniques play a significant role in the workflow of genomics researchers, and they are regularly employed in the interpretation of genomics data. However, the vast majority of genomics researchers have little or no formal training in data visualization design. Therefore, they require guidance on designing effective visualizations for a given set of data and analysis tasks. In this work, I present a knowledge-based recommendation system for genomics visualization. The system allows genomics researchers to navigate through a selection of visualization options and identify the techniques that meet their preferences and requirements. The first step in building the recommendation system was to identify the data structures, analytical tasks, and visualization designs used in genomics analysis. The required information was gathered from the survey paper by Nusrat et al. [35], where the authors contributed a data, task, and visualization taxonomy for genomics visualization. Next, I characterized the typical design workflow of a genomics visualization. As shown in Fig. 8 (B), to create a genomics visualization, a designer needs to make several design decisions like the choice of marks and channels to encode genomics data or how to layout the marks and channels. Our analysis found that design stages are sequential, meaning each step feeds into the next one. For instance, the choice of alignment depends on the choice of layout. The third step in creating a recommendation model was identifying design guidelines that inform the selection of a visualization. Design guidelines for our recommendation models are derived from general visualization graphical, and perception studies and analysis of genomics visualization literature published at visualization conferences.

Input and Output of the Recommendation System: Both data features and types of analytical tasks are inputs to the recommendation model. Data information is either provided explicitly by users or collected automatically from the standard file formats used for genomics data. In addition to data-driven recommendations, our system supports task-based recommendations. Unlike data descriptions, tasks that users are intended to perform are difficult to infer, which requires task descriptions to be explicitly specified by users. The recommendation system's output includes a custom interactive visualization implementation and a list of existing genomics visualization tools and libraries that match the user's data and task requirements.

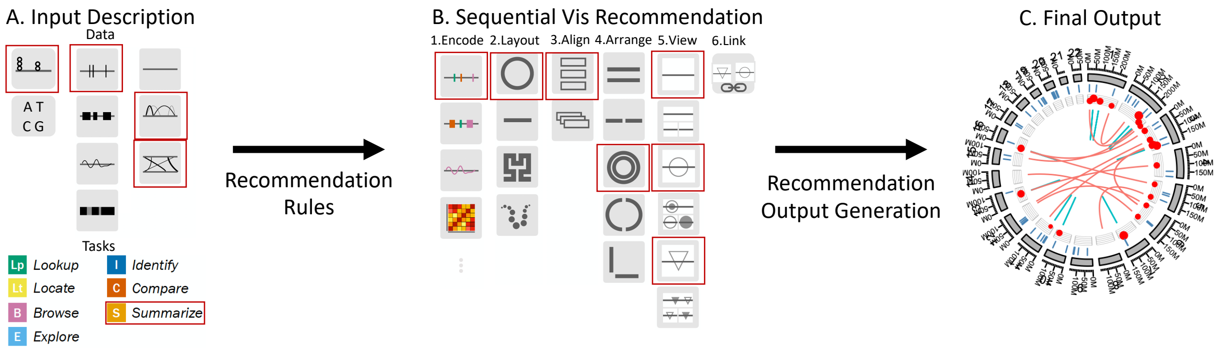


Figure 8: A schematic representation of the three stages of a visualization recommendation system for genomics data. The input visually presents the data and task specification for the system. Based on data and task requirements, the sequential recommendation model identifies suitable visual encoding at each step of the visualization grammar, and the output visualization represents the final deliverable to the user for their analysis.

Status: This work was accepted as a poster at IEEE Vis 2020 [39] and received the "Best Poster Award" at the conference. The poster gave an overview of the system design and explained in detail the recommendation model. I plan to submit this work as a full paper to IEEE Vis 2021. For the submission, I am working on developing the front-end of the recommendation system. The recommendation system's front-end will allow genomics researchers to provide their data and task specification as input to the recommendation model. The front-end will also enable the users to analyze the recommended genomics visualization. After completing the front-end development, I will evaluate the effectiveness and expressivity of the recommendation model output. To assess the effectiveness, I will carry out a case-study based evaluation with domain experts. In the assessment, domain experts will use the recommendation system and evaluate the visualization output's accuracy. I will also observe if the recommendation system helped the domain experts find an accurate but unexpected visualization output. This question will help us evaluate the serendipity and pedagogical factors of the system. To assess the expressivity, I will try creating the common genomics visualization identified in the survey paper by Nusrat et al. [35]. If the recommendation system covers many visualization techniques from the survey, it will indicate high expressivity of the recommendation system.

8.2 Recommendation System for Tree Visualization

Motivation & Preliminary Results: In Sec. 7 (Tree Visualizations), I discussed our plan for developing task-based design guidelines for tree visualizations. The design guidelines are theoretical and will be communicated to visualization creators in the 'printed on paper' form as a research paper. However, guidelines available in research form may not be easy to access or apply, see Sec. 1. Therefore, through my work, I want to make the guidelines actionable and usable by visualization creators. To make the design guidelines actionable and easily applicable, I will create a system to recommend a tree visual encoding based on a user's data and tasks. My ongoing research work on the genomics visualization recommendation system is actively helping me learn valuable system design and engineering implications for the new proposed work, including user interface designs for the recommendation tool, and techniques to decouple the recommendation engine or model from the interface to increase the portability of the recommendation engine into different programming languages such as Python and Javascript. I plan to use the skills of designing a recommendation system from genomics visualization research and combine it with the theoretical design guidelines of creating tree visualizations identified in Sec. 7 (Tree Visualizations) to develop a tree visualization recommendation system.

Research Plan: The core of recommendation system is the recommendation model. The recommendation model will contain rules to support mapping of data and task requirements to appropriate visualization encodings. The rules will be generated from design guidelines discussed in Sec. 7. The recommendation model will take in the tree data type and the user's already abstracted tasks and then generate a ranked list of supporting tree visualization encoding techniques with the help of visualization design base. For example, if node data is "aggregatable" and the task is to "identify outliers", then use a "treemap" visualization. The recommendation rules can be stored externally in a text file. Storing recommendation rules externally will allow easy update of the rules in the future. For the recommendation, I will also implement an algorithm to apply the rules and generate a ranked list of visualization recommendation. The recommendation algorithm will be implemented in Javascript to ensure seamless integration with the web-based recommendation tool. The recommendation system will be implemented as a system with a website front-end UI to allow a user to input their tree data structure and abstract tasks and get out a series of visual encoding recommendations. We will provide an interactive user interface to help users determine their data and task requirements and select them in a step-by-step guided input to the recommendation system.

Expected Outcomes: Accomplishment of this project will yield a new recommendation system built on the design guidelines derived from visualization evaluation papers, to recommend tree visual encoding techniques. The underlying code of the recommendation system, including the underlying algorithm, will be made open source and publicly available with supporting documentation. The system will be implemented with a front-end website to serve as the user interface. This system will help researchers, designers, and students alike be able to more easily navigate the vast tree visualization space, create tree visualizations, reduce implicit bias in the visualization design process, and support visualization education.

Validation Plan: I will conduct a user study with quantitative and qualitative metrics to measure confidence, trust, novelty, and serendipity with use of the new system. The new recommendation system will be evaluated and validated by visualization practitioners and researchers. I identify visualization practitioners and researchers as individuals who are familiar with visualization design best practices through formal training or work experience. We need participants with visualization training or knowledge because the recommendation system is primarily designed to help them in the task of identifying appropriate tree visualization encodings. Since the system is still under research stage, the exact plan of the evaluation will be determined closer to the study. However, the broad goal of the evaluation will be to understand if the output of the recommendation system is accurate and matches expectations of visualization creators. Furthermore, I will also like to study if the recommendation system can help visualization creators find visualization techniques they were not previously familiar with but the recommendation system helped them find it. This will be an online study without any in-person component.

9 Conclusion

This thesis presents theoretical knowledge and practical tools to improve visualization design for practitioners and researchers. I use the case of tree visualizations as the overarching theme to identify several major shortcomings of our theoretical knowledge about visualization design and propose solutions to overcome the existing challenges. Till this point in my research, I have solved novel visualization design problems, and in the process, identified challenges with the existing theoretical resources for designing tree visualizations. I have used these challenges to improve the theoretical support to abstract tree visualization tasks, which ultimately support the selection of accurate tree visualization encoding. Currently, I am working on using the new tree visualization theory and a survey of tree visualization studies to build tree visualization design guidelines. I plan to finish the development of guidelines by the Summer of 2021. With the new tree visualization design guidelines, I plan to develop a system for tree visualization recommendation. As the design guidelines are a prerequisite to the recommendation system, I anticipate to start the work on the tree visualization recommendation system in the Fall of 2021 and finish the work in the Spring of 2022. My work on tree visualizations has constantly drawn support from my other research projects, which also fall in the general theme of improving the visualization design pipeline. In my thesis, I present task-based visualization design guidelines for data glyphs and timelines. The design guidelines project inspired me to survey for task-based guidelines for tree visualizations. In the survey, I found that tree visualization design guidelines are not well-curated and need additional work. I also anticipate my genomics visualization recommendation system will help me design and develop the tree visualization recommendation system.

Motivated by the goal to improve the visualization design pipeline, I will make the following contributions. Through a series of practical visualization design studies, I identify the existing challenges of creating visualizations. The design studies also solve visualization design problems in the critical fields of medical diagnosis and cybersecurity. Next, I contribute theory to abstract tree visualization tasks, with a dataset for visualization creators to analyze tasks and their abstraction of over 200 tasks. I also contribute task-based design guidelines for tree visualizations, data glyphs, and timelines. Finally, to apply visualization design guidelines in a practical context, I develop visualization recommendation systems. These contributions help the visualization community understand the existing problems with the visualization design pipeline and contribute knowledge and resources to eliminate the challenges.

10 Appendix

As appendix to the thesis proposal document, I have attached the author's copy of the articles that have either been published or are in preparation for submission to reputable journals, conferences, workshops, and poster sessions. A complete list of the articles with their publication venue is listed below.

10.1 Journal and Conference Papers

- A. Pandey et al.,"A State-of-the-Art Survey of Tasks for Tree Design and Evaluation with a Curated Task Dataset," to Appear in IEEE Transactions on Visualization and Computer Graphics. (Conditionally accepted for publication)
- S. D. Bartolomeo, **A. Pandey**, A. Leventidis, D. Saffo, U. H. Syeda, E. Carstensdottir, M. S. El-Nasr, M. A. Borkin, and C. Dunne, "Evaluating the Effect of Timeline Shape on Visualization Task Performance", in Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems CHI '20. Association for Computing Machinery. DOI:https://doi.org/10.1145/3313831.3376237
- A. Pandey et al., "CerebroVis: Designing an Abstract yet Spatially Contextualized Cerebral Artery Network Visualization," in IEEE Transactions on Visualization and Computer Graphics, vol. 26, no. 1, pp. 938-948, Jan. 2020, doi: 10.1109/TVCG.2019.2934402
- A. Pandey et al., "Effect of Anthropomorphic Glyph Design on Categorization Accuracy", in-prep for submission

10.2 Workshop Papers

- A. Pandey, U. H. Syeda and M. A. Borkin, "Towards Identification and Mitigation of Task-Based Challenges in Comparative Visualization Studies," 2020 IEEE Workshop on Evaluation and Beyond - Methodological Approaches to Visualization (BELIV), Salt Lake City, UT, USA, 2020, pp. 55-64, doi: 10.1109/BELIV51497.2020.00014
- A. Pandey, Y. Zhang, J. A. Guerra-Gomez, A. G. Parker, and M. A. Borkin, "Digital Collaborator: Augmenting Task Abstraction in Visualization Design with Artificial Intelligence", 2020, accepted at CHI 2020 workshop

10.3 Posters

- A. Pandey et al., "Towards a Knowledge-Based Recommendation System for Genomics Visualization.", accepted at IEEE Vis 2020
- A. Pandey et al., "Segmentrix: A Network Visualization Tool to Develop and Monitor Micro-Segmentation Strategies", accepted at Vizsec 2019

References

[1] Bilal Alsallakh et al. "Visualizing Sets and Set-typed Data: State-of-the-Art and Future Challenges". In: *EuroVis - STARs*. Ed. by R. Borgo, R. Maciejewski, and I. Viola. The Eurographics Association, 2014. ISBN: -. DOI: 10.2312/eurovisstar.20141170.

[2] R. Amar, J. Eagan, and J. Stasko. "Low-level components of analytic activity in information visualization". In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* Oct. 2005, pp. 111–117. DOI: 10.1109/INFVIS.2005.1532136.

[3] Gennady Andrienko et al. "A conceptual framework and taxonomy of techniques for analyzing movement". In: *Journal of Visual Languages & Computing* 22.3 (2011), pp. 213–232. DOI: 10.1109/VL.1996.545307.

[4] Michelle A. Borkin et al. "What Makes a Visualization Memorable?" In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2306–2315. ISSN: 1077-2626. DOI: 10.1109/TVCG.2013.234. URL: http://dx.doi.org/10.1109/TVCG.2013.234.

[5] M. Bostock, V. Ogievetsky, and J. Heer. "D³ Data-Driven Documents". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (Dec. 2011), pp. 2301–2309. ISSN: 1941-0506. DOI: 10.1109/TVCG.2011.185.

[6] M. Brehmer and T. Munzner. "A Multi-Level Typology of Abstract Visualization Tasks". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2376–2385. ISSN: 2160-9306. DOI: 10.1109/TVCG.2013.124.

[7] M. Brehmer et al. "Timelines Revisited: A Design Space and Considerations for Expressive Storytelling". In: *IEEE Transactions on Visualization and Computer Graphics* 23.9 (2017), pp. 2151–2164. ISSN: 1077-2626. DOI: 10.1109/TVCG.2016.2614803.

[8] M. Burch et al. "Evaluation of Traditional, Orthogonal, and Radial Tree Diagrams by an Eye Tracking Study". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2440–2448.

[9] N. Cawthon and A. V. Moere. "The Effect of Aesthetic on the Usability of Data Visualization". In: *2007 11th International Conference Information Visualization (IV '07)*. 2007, pp. 637–648. DOI: 10.1109/IV.2007.147.

[10] M. Chen et al. "Pathways for Theoretical Advances in Visualization". In: *IEEE Computer Graphics and Applications* 37.4 (2017), pp. 103–112. ISSN: 1558-1756. DOI: 10.1109/MCG.2017.3271463.

[11] William S. Cleveland and Robert McGill. "Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods". In: *Journal of the American Statistical Association* 79.387 (1984), pp. 531–554. ISSN: 01621459. URL: http://www.jstor.org/stable/2288400.

[12] Sara Di Bartolomeo et al. "Evaluating the Effect of Timeline Shape on Visualization Task Performance". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–12. ISBN: 9781450367080. DOI: 10.1145/3313831.3376237. URL: https://doi.org/10.1145/3313831.3376237.

[13] A. Diehl et al. "Visguides: A Forum for Discussing Visualization Guidelines". In: *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers*. EuroVis '18. Brno, Czech Republic: Eurographics Association, 2018, pp. 61–65.

[14] Elliot K Fishman et al. "Volume Rendering versus Maximum Intensity Projection in CT Angiography: What Works Best, When, and Why". In: *Radiographics* (2006). DOI: https://doi.org/10.1148/rg.263055186.

[15] *FLOWINGDATA, url = https://flowingdata.com/about/*.

[16] *From data to Viz, url = https://www.data-to-viz.com/*.

[17] J. Fuchs et al. "A Systematic Review of Experimental Studies on Data Glyphs". In: *IEEE Transactions on Visualization and Computer Graphics* 23.7 (July 2017), pp. 1863–1879. ISSN: 1941-0506. DOI: 10.1109/TVCG.2016.2549018.

[18] Steve Haroz, Robert Kosara, and Steven L. Franconeri. "ISOTYPE Visualization: Working Memory, Performance, and Engagement with Pictographs". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: Association for Computing Machinery, 2015, pp. 1191–1200. ISBN: 9781450331456. DOI: 10.1145/2702123.2702275. URL: https://doi.org/10.1145/2702123.2702275.

[19] Kevin Hu et al. "VizML: A Machine Learning Approach to Visualization Recommendation". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–12. ISBN: 9781450359702. DOI: 10.1145/3290605.3300358. URL: https://doi.org/10.1145/3290605.3300358.

[20] Phillip Isola et al. "Understanding the Intrinsic Memorability of Images". In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor et al. Curran Associates, Inc., 2011, pp. 2429–2437. URL: http://papers.nips.cc/paper/4451-understanding-the-intrinsic-memorability-of-images.pdf.

[21] B. Johnson and B. Shneiderman. "Tree-maps: a space-filling approach to the visualization of hierarchical information structures". In: *Proceeding Visualization '91*. Oct. 1991, pp. 284–291. DOI: 10.1109/VISUAL.1991.175815.

[22] Pawandeep Kaur and Michael Owonibi. "A Review on Visualization Recommendation Strategies". In: Feb. 2017. DOI: 10.5220/0006175002660273.

[23] Stephan Kerpedjiev et al. "AutoBrief: A Multimedia Presentation System for Assisting Data Analysis". In: *Comput. Stand. Interfaces* 18.6–7 (Dec. 1997), pp. 583–593. ISSN: 0920-5489. URL: https://doi.org/10.1016/S0920-5489(97)00022-6.

[24] N. Kerracher, J. Kennedy, and K. Chalmers. "A Task Taxonomy for Temporal Graph Visualisation". In: *IEEE Transactions on Visualization and Computer Graphics* 21.10 (Oct. 2015), pp. 1160–1172. ISSN: 1941-0506. DOI: 10.1109/TVCG.2015.2424889.

[25] David M. Klumpar, Kevin Anderson, and Avangelos Simoudis. "RAVE: Rapid Visualization Environment". In: 1994.

[26] A. Kobsa. "User Experiments with Tree Visualization Systems". In: *IEEE Symposium on Information Visualization*. Oct. 2004, pp. 9–16. DOI: 10.1109/INFVIS.2004.70.

[27] Bongshin Lee et al. "Task Taxonomy for Graph Visualization". In: *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*. BELIV '06. Venice, Italy: Association for Computing Machinery, 2006, pp. 1–5. ISBN: 1595935622. DOI: 10.1145/1168149.1168168.

[28] Manuel Lima. *The book of trees: visualizing branches of knowledge*. Princeton Architectural Press, 2014.

[29] Jock Mackinlay. "Automating the Design of Graphical Presentations of Relational Information". In: *ACM Trans. Graph.* 5.2 (Apr. 1986), pp. 110–141. ISSN: 0730-0301. DOI: 10.1145/22949.22950. URL: https://doi.org/10.1145/22949.22950.

[30] M. Meyer, T. Munzner, and H. Pfister. "MizBee: A Multiscale Synteny Browser". In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (Nov. 2009), pp. 897–904. ISSN: 1941-0506. DOI: 10.1109/TVCG.2009.167.

[31] D. Moritz et al. "Formalizing Visualization Design Knowledge as Constraints: Actionable and Extensible Models in Draco". In: *IEEE Transactions on Visualization and Computer Graphics* 25.1 (Jan. 2019), pp. 438–448. ISSN: 1941-0506. DOI: 10.1109/TVCG.2018.2865240.

[32] T. Munzner. "A Nested Model for Visualization Design and Validation". In: *IEEE Transactions on Visualization and Computer Graphics* 15.6 (Nov. 2009), pp. 921–928. ISSN: 1941-0506. DOI: 10.1109/TVCG.2009.111.

[33] Tamara Munzner. *Visualization analysis and design*. CRC press, 2014.

[34] C. Nobre et al. "The State of the Art in Visualizing Multivariate Networks". In: *Computer Graphics Forum* 38.3 (2019), pp. 807–832. DOI: https://doi.org/10.1111/cgf.13728. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13728. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13728.

[35] S. Nusrat, T. Harbig, and N. Gehlenborg. "Tasks, Techniques, and Tools for Genomic Data Visualization". In: *CGF* (). DOI: 10.1111/cgf.13727. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13727. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13727.

[36] Sabrina Nusrat and Stephen Kobourov. "Task Taxonomy for Cartograms". In: *Eurographics Conference on Visualization (EuroVis) - Short Papers*. Ed. by E. Bertini, J. Kennedy, and E. Puppo. The Eurographics Association, 2015. DOI: 10.2312/eurovisshort.20151126.

[37] A. Pandey, U. H. Syeda, and M. A. Borkin. "Towards Identification and Mitigation of Task-Based Challenges in Comparative Visualization Studies". In: *2020 IEEE Workshop on Evaluation and Beyond - Methodological Approaches to Visualization (BELIV)*. 2020, pp. 55–64. DOI: 10.1109/BELIV51497.2020.00014.

[38] A. Pandey et al. "CerebroVis: Designing an Abstract yet Spatially Contextualized Cerebral Artery Network Visualization". In: *IEEE Transactions on Visualization and Computer Graphics* 26.1 (Jan. 2020), pp. 938–948. ISSN: 1941-0506. DOI: 10.1109/TVCG.2019.2934402.

[39] Aditeya Pandey, Sehi L'Yi, and Nils Gehlenborg. "Towards a Knowledge-Based Recommendation System for Genomics Visualization". In: ().

[40] Aditeya Pandey et al. "Poster: Segmentrix: A Network Visualization Tool to Develop and Monitor Micro-Segmentation Strategies". In: ().

[41] *Power BI, url = https://powerbi.microsoft.com/en-us/.*

[42] Steven F. Roth and Joe Mattis. "Data Characterization for Intelligent Graphics Presentation". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '90. Seattle, Washington, USA: Association for Computing Machinery, 1990, pp. 193–200. ISBN: 0201509326. DOI: 10.1145/97243.97273. URL: https://doi.org/10.1145/97243.97273.

[43] H. Schulz. "Treevis.net: A Tree Visualization Reference". In: *IEEE Computer Graphics and Applications* 31.6 (Nov. 2011), pp. 11–15. ISSN: 1558-1756. DOI: 10.1109/MCG.2011.103.

[44] H. Schulz et al. "A Design Space of Visualization Tasks". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (Dec. 2013), pp. 2366–2375. ISSN: 2160-9306. DOI: 10.1109/TVCG.2013.120.

[45] M. Sedlmair, M. Meyer, and T. Munzner. "Design Study Methodology: Reflections from the Trenches and the Stacks". In: *IEEE Transactions on Visualization and Computer Graphics* 18.12 (Dec. 2012), pp. 2431–2440. ISSN: 1941-0506. DOI: 10.1109/TVCG.2012.213.

[46] B. Shneiderman. "The eyes have it: a task by data type taxonomy for information visualizations". In: *Proceedings 1996 IEEE Symposium on Visual Languages*. 1996, pp. 336–343.

[47] John Stasko et al. "An evaluation of space-filling information visualizations for depicting hierarchical structures". In: *International journal of human-computer studies* 53.5 (2000), pp. 663–694. DOI: https://doi.org/10.1006/ijhc.2000.0420.

[48] Chris Stolte, Diane Tang, and Pat Hanrahan. "Polaris: A System for Query, Analysis, and Visualization of Multidimensional Databases". In: *Commun. ACM* 51.11 (Nov. 2008), pp. 75–84. ISSN: 0001-0782. DOI: 10.1145/1400214.1400234. URL: https://doi.org/10.1145/1400214.1400234.

[49] Uzma Haque Syeda et al. "Design Study "Lite" Methodology: Expediting Design Studies and Enabling the Synergy of Visualization Pedagogy and Social Good". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–13. ISBN: 9781450367080. DOI: 10.1145/3313831.3376829. URL: https://doi.org/10.1145/3313831.3376829.

[50] Danielle Albers Szafir. "The Good, the Bad, and the Biased: Five Ways Visualizations Can Mislead (and How to Fix Them)". In: *Interactions* 25.4 (June 2018), pp. 26–33. ISSN: 1072-5520. DOI: 10.1145/3231772. URL: https://doi.org/10.1145/3231772.

[51] *Tableau Software, url = https://www.tableau.com/.*

[52] Manasi Vartak et al. "SeeDB: Efficient Data-Driven Visualization Recommendations to Support Visual Analytics". In: *Proc. VLDB Endow.* 8.13 (Sept. 2015), pp. 2182–2193. ISSN: 2150-8097. DOI: 10.14778/2831360.2831371. URL: https://doi.org/10.14778/2831360.2831371.

[53] Manasi Vartak et al. "Towards Visualization Recommendation Systems". In: *SIGMOD Rec.* 45.4 (May 2017), pp. 34–39. ISSN: 0163-5808. DOI: 10.1145/3092931.3092937. URL: https://doi.org/10.1145/3092931.3092937.

[54] Martin Voigt, Stefan Pietschmann, and Lars Grammel. "Context-aware Recommendation of Visualization Components". In: 2012.

[55] K. Wongsuphasawat et al. "Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations". In: *IEEE Transactions on Visualization and Computer Graphics* 22.1 (Jan. 2016), pp. 649–658. ISSN: 1941-0506. DOI: 10.1109/TVCG.2015.2467191.

Appendix

Below, we have author copies of all the published and completed papers and posters.

Papers

- A. Pandey et al.,“A State-of-the-Art Survey of Tasks for Tree Design and Evaluation with a CuratedTask Dataset,” to Appear in IEEE Transactions on Visualization and Computer Graphics. (Conditionally accepted for publication)

- S. D. Bartolomeo,A. Pandey, A. Leventidis, D. Saffo, U. H. Syeda, E. Carstensdottir, M. S. El-Nasr, M. A. Borkin, and C. Dunne, “Evaluating the Effect of Timeline Shape on Visualization Task Performance”, in Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems CHI ’20. Association for Computing Machinery. DOI: https://doi.org/10.1145/3313831.3376237

- A. Pandey et al., “CerebroVis: Designing an Abstract yet Spatially Contextualized Cerebral ArteryNetwork Visualization,” in IEEE Transactions on Visualization and Computer Graphics, vol. 26, no.1, pp. 938-948, Jan. 2020, doi: 10.1109/TVCG.2019.2934402

- A. Pandey et al., “Effect of Anthropomorphic Glyph Design on Categorization Accuracy”, in-prep for submission

Workshop Papers

- A. Pandey, U. H. Syeda and M. A. Borkin, “Towards Identification and Mitigation of Task-Based Challenges in Comparative Visualization Studies,” 2020 IEEE Workshop on Evaluation and Beyond Methodological Approaches to Visualization (BELIV), Salt Lake City, UT, USA, 2020, pp. 55-64, doi:10.1109/BELIV51497.2020.00014

- A. Pandey, Y. Zhang, J. A. Guerra-Gomez, A. G. Parker, and M. A. Borkin, “Digital Collaborator: Augmenting Task Abstraction in Visualization Design with Artificial Intelligence”, 2020, accepted at CHI 2020 workshop

Posters

- A. Pandey et al., “Towards a Knowledge-Based Recommendation System for Genomics Visualization”, accepted at IEEE Vis 2020

- A. Pandey et al., “Segmentrix: A Network Visualization Tool to Develop and Monitor Micro-Segmentation Strategies”, accepted at Vizsec 2019

A State-of-the-Art Survey of Tasks for Tree Design and Evaluation with a Curated Task Dataset

Aditeya Pandey, Uzma Haque Syeda, Chaitya Shah, John A. Guerra-Gomez, and Michelle A. Borkin

Abstract—In the field of information visualization, the concept of "tasks" is an essential component of theories and methodologies for how a visualization researcher or a practitioner understands what tasks a user needs to perform and how to approach the creation of a new design. In this paper, we focus on the collection of tasks for tree visualizations, a common visual encoding in many domains ranging from biology to computer science to geography. In spite of their commonality, no prior efforts exist to collect and abstractly define tree visualization tasks. We present a literature review of tree visualization papers and generate a curated dataset of over 200 tasks. To enable effective task abstraction for trees, we also contribute a novel extension of the Multi-Level Task Typology to include more specificity to support tree-specific tasks as well as a systematic procedure to conduct task abstractions for tree visualizations. All tasks in the dataset were abstracted with the novel typology extension and analyzed to gain a better understanding of the state of tree visualizations . These abstracted tasks can benefit visualization researchers and practitioners as they design evaluation studies or compare their analytical tasks with ones previously studied in the literature to make informed decisions about their design. We also reflect on our novel methodology and advocate more broadly for the creation of task-based knowledge repositories for different types of visualizations. The Supplemental Material will be maintained on OSF: https://osf.io/u5ehs/

Index Terms—STAR, Tree, Tasks, Task Abstraction, Theory, Datasets

1 Introduction

In information visualization, analytical tasks refer to actionable and measurable user goals that aid with visualization design and evaluation [1]. Tasks help visualization practitioners understand the user's needs and make informed design choices [1], [2], [3]. With clear tasks, a supportive and effective visualization can be created. For example, a financial analyst may want to analyze performance of stocks in different sectors of the market, particularly in the aftermath of a major global crisis to identify outliers that survived recessions and performed well despite an economic slowdown. In another instance, an epidemiologist may want to analyze the patterns in the branching of a virus strain and compare how different strains evolve over time. The task to be accomplished by both the financial analyst and the epidemiologist relate to the visualization task of analyzing hierarchical information, and thus informs the visualization creator that the users require a tree visualization to address their goals. Beyond support for visualization design, task knowledge also plays an essential role in the evaluation of visualizations. Evaluations, including those in design studies, rely on visualization tasks to measure the effectiveness of various aspects of visualization tools such as the visual encoding [4], interaction [5], and user experience [6]. Because tasks serve such a critical role in visualization design and evaluation, the visualization community has created systematic organizations of task information in the form of taxonomies [7], [8], [9], [10], [11], [12], surveys [13], [14], [15], and frameworks [1], [16], [17], [18]. All these task resources collectively aspire to promote a better organization and understanding of analytical tasks for visualization encodings.

Although there are many general task taxonomies in visualization [1], [11], [12], these frameworks sometimes lack the specificity to support task abstractions for specific dataset types such as temporal, spatial-temporal, networks, and trees. Specialized task taxonomies and frameworks have been developed for some of these under-supported data types including networks [7] and spatial-temporal data [17]. Trees, however, do not have a specialized taxonomy to address their unique characteristics. Tree visualizations of hierarchical data are common in many fields such as software engineering, machine learning, geography, finance, and biology. The common applications of tree visualizations in these fields are organization and representation of code-bases in software engineering, explainability of decision-tree models in machine learning, presentation of natural geographical phenomenon like river branching in geography, and exploration of genetic evolution data in biology [1], [19]. The lack of a formal task abstraction framework for trees makes it more challenging for visualization creators to effectively design and evaluate tree visualizations.

Existing general frameworks such as the Multi-Level Task Typology (MLTT) [1], [2], offers high-level support for the abstraction of tree visualization tasks. The typology supports abstraction of tree "targets" into "topology", "path", and "attribute". In many cases, high-level targets can lead to an ambiguous task description. For instance, applying the MLTT framework for the abstraction of the previously discussed examples of a financial analyst interested in finding the outlier market sectors and an epidemiologist interested in comparing the different strains of virus yields the same abstraction result: the goal of the task is to perform "lookup" on the "topology". However, in practice the two tasks are fundamentally different: the financial analyst is interested in "looking up" the "ancestor" nodes of outlier stocks in the stock market tree, and the epidemiologist wants to "compare" properties like "height" and "fanout" of different branches in the tree. The high-level tree-specific targets in the MLTT specification hides these nuances in the abstract task descriptions. To address this problem, we contribute an extension of the MLTT framework which adds more specificity for trees to the task abstraction definition based on an extensive literature review and survey of tasks described in the tree visualization literature.

• *Aditeya Pandey, Uzma Syeda Haque, Chaitya Shah, John A. Guerra-Gomez, and Michelle A. Borkin are with Northeastern University. E-mail: {pandey.ad,syeda.u,shah.cha,jguerra, m.borkin}@northeastern.edu*

Manuscript received April 19, 2005; revised August 26, 2015.

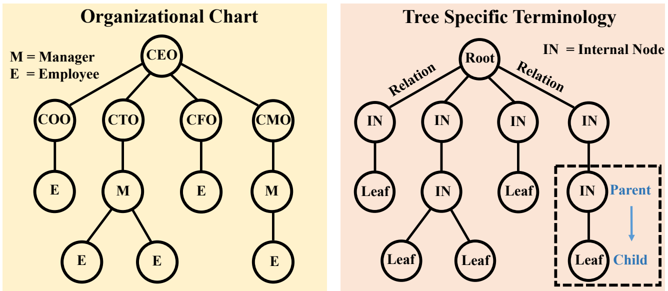


Fig. 1. Left: An organization chart as a tree visualization that shows the structure of an organization and the relationships and relative ranks of its parts and positions/jobs. Right: In the graph theory literature, the nodes and links of trees are identified by a special set of terms based on their position in the hierarchy. This figure visually represents the common tree terminologies.

For the survey, we reviewed over 1000 relevant tree visualization papers and identified 54 papers with tasks. Based on these papers we built a tree task dataset with a collection of 212 tasks. Each of these tasks was abstracted using our novel MLTT extension for trees in order to enable creators of tree visualizations to find similarities and differences between their abstract tasks with that of others in the dataset. Based on the survey and tasks, we also contribute a discussion of opportunities for research in the community as well as the advocation of task-collections for other visual encodings. In order to provide transparency and support reproducibility, we provide a detailed step by step procedure on how to systematically perform the task abstraction. In addition, to communicate our survey results to the visualization research and design community, we contribute an interactive public-facing website [1]. The website includes a meta-analysis of the surveyed papers, a summary of tasks and visual encodings found in each surveyed paper, an interactive interface to explore a novel task dataset, and an exploration section to identify potential areas of future work in terms of evaluation or design of tree visualizations.

Contributions: We contribute a state-of-the-art survey of tree visualization tasks and a systematic method to identify the corresponding analytical tasks using a novel task typology for trees extended from the more general MLTT framework. The results of this task survey are curated as a novel dataset. We also contribute a meta-analysis of the task dataset and identify areas of opportunity for future visualization research.

2 Background & Related Works

In this section, we introduce the MLTT framework and its significance on the abstraction of tasks. We provide a reference for common tree concepts and terminologies, with an overview of different tree visualization techniques. Finally, we discuss the scope of our survey and where our work fits in the space of tree visualization task theory.

2.1 Summary of the Multi-Level Task Typology

Tasks in information visualization literature are often expressed in domain-specific terminology. For example, in a financial domain, an analyst may want to identify stocks that survived recession, or in epidemiology, a scientist may want to know how strains of a virus evolve. However, it is not easy to compare different visualization tasks across applications based on their domain-specific language. Task abstraction is the process of removing domain-specific terminology from the task description to promote easy understanding and adoption of the task-based results in application domains that are not directly related to the research problem [3]. As discussed in Sec. 1, there are many general frameworks to abstract a visualization task [1], [11], [12]. In this paper, we chose to use and extend the Multi-Level Task Typology (MLTT) framework by Brehmer & Munzner [1], [2] because it is the only framework to date that discusses tasks in the context of data and visual encoding, with a special focus on both the actions and targets associated with a visualization task. The MLTT framework is motivated by three questions: **what** data is being visualized including identification of the dataset type (tree, network, table, spatial data, etc), the data types (nodes, links, attributes), and the attribute types (categorical, ordinal and quantitative); **why** was the visualization created, or what are the intended tasks (as a combination of action and targets) users should be able to perform on it; and **how** visualization is represented in terms of marks, channels and interactions used. In our work, we apply and extend the "why" part of the framework for categorizing and abstracting tree visualization tasks.

The "why" part of the MLTT framework helps the user understand why a particular task is carried out and breaks down the task into high-, mid-, and low-level categorization along with the final target of the task. In the framework, each categorization consists of abstract concepts to delineate the various objectives at each stage of the task. For instance, the high-level categorizations analyze whether the visualization is used to *consume (discover, present, and enjoy)* or *produce (annotate, record, and derive)* data. The mid-level actions *(lookup, locate, browse, and explore)* describe the type of *search* carried out based on the target and location knowledge. The low-level actions *(identify, compare, and summarize)* represent the type of *query* performed on the target. Targets can be different kinds of *data (e.g., trends, or outliers)*, *attributes (e.g., extremum)*, and *topology (for network data)*. Although the framework allows users to abstract domain-specific visualization tasks with the help of multi-level classification and the use of descriptive abstract concepts, the framework does not claim to be an exhaustive visualization task taxonomy. We used each of these levels of "why" to better understand how people use tree visualizations and added an extension to the framework's target part to abstract tree visualization tasks as discussed in Sec. 4.

2.2 Trees

Tree visualization is a topic broadly covered in the literature, both in information visualization and other domains (e.g., [20], [21], [22], [23]). However, it is common to find a large variation in how people address trees in their respective domains and fields. For example, biologists commonly work with phylogenetic trees [23], which only contain attributes in the leaf nodes and are usually represented using node-link visualizations such as dendrograms. On the other hand, financial analysts working on the stock market will commonly use a Treemap [21], as the market caps can be aggregated to represent the value of a sector. A dendrogram could be used to represent the map of the market, but since the phylogenetic tree doesn't have inner numeric values, it cannot be represented with a Treemap. Despite this, the literature will use the term tree to refer to the dataset and visualization on both scenarios, as if they were interchangeable. Because of this, and in order to perform a proper classification of the papers in this survey, we present a reference definition of the concept of a tree and its associated terminologies in the context of information visualization.

1. https://intervis-projects.ccs.neu.edu/Tree-Visualization-Survey/

Attribute	Attribute Of	Data Type	Definition	Also Known As
Height	Tree, Subtree	Integer	The height of a tree is the length of the longest path from the root.	-
Size	Tree, Subtree	Integer	The size of a tree is the count of all the nodes in a tree.	-
Fanout	Tree, Subtree	Integer	The width of a tree is the maximum number of nodes in a level.	Width, Breadth
Balance	Tree, Subtree	Boolean	A tree is height-balanced if the depth of any two leaf nodes differs by at most 1.	-
Order	Tree, Subtree	Boolean	An ordered tree is a rooted tree in which each node's children are assigned a fixed ordering.	-
Ancestor/Descendant	Node, Subtree	String	A vertex w is called a descendant of a vertex v (and v is called an ancestor of w), if v is on the unique path from the root to w.	-
Depth	Node, Subtree	Integer	The depth of a node is the distance of the node from the root of the tree.	Level
Degree	Node	Integer	The degree of node is the count of immediate child nodes.	Fanout
Siblings	Node	String	Nodes that have the same parent are known as siblings – siblings are, by definition, always on the same level.	-

TABLE 1
In this table, we list the structural attributes of a tree. For each attribute, we provide information like its association with the tree visualization ("Attribute Of"), the type of attribute ("Data Type") of the attribute, a one-line definition ("Definition"), and commonly known aliases ("Also Known as"). The structural attributes are visually illustrated in Fig. 4 (Targets).

2.2.1 *Tree Definition and Terminology*

A *tree* is defined as a collection of nodes and links. A *node* is a data structure that can have an identifier (id) and a value. A *link* in a tree is a data structure that connects two *unique* nodes. Similar to the node, a link can also have values associated with them. A key aspect that differentiates trees in graph theory is the "hierarchical" relationship that exists within the nodes. Hierarchical relations categorize nodes in a tree dataset as "above" or "parent", "below" or "child", and "at the same level" or "sibling".

Tree terminology plays an essential role in the analysis and discussion of tree visualization tasks. Given the variation in vocabulary, it is essential to revisit the tree visualization terminology and clarify our interpretation of these essential terms. Moreover, we discuss the definitions in terms of tasks in order to facilitate the discussion of tree-specific tasks in Sec. 4.

Tree: A tree is a collection of *all* nodes and links in a tree dataset. We classify a task as a tree-level task when its focus is the entire tree visualization. For example, in Fig. 1, the Organization Chart is a tree.

Node: A node is a *singular* unit of a tree visualization. We classify a task as a node-level task when its focus is to analyze a single node of the tree without additional hierarchical contexts like the parent, child, or sibling nodes. For example, in Fig. 1 (Organization Chart), a node can be any employee of the organization.

Subtree: A subtree is a *subset* of nodes and links in a tree dataset that preserves all properties of a tree. Another way to define a subtree is in terms of the node: any node in the tree dataset, along with all its descendants, forms a subtree. We classify a task as a subtree-level task when its focus is on a subset of nodes and links rather than the entire tree dataset. For example, in Fig. 1 (Organization Chart), the CTO and all the employees reporting to the CTO form a subtree.

In addition to the above discussed definitions, trees also have special terminologies for nodes at different levels in the hierarchy. In a tree, any node that has a child node is called an *internal node*. A special type of internal node is the *root node*. The root node is the start of the hierarchy and has no parent node. The nodes at the bottom of the hierarchy without any child nodes are called *external or leaf nodes*. In Fig. 1 (Organization Chart), the CEO is the root node, the employees are the leaf nodes, and all other nodes are internal nodes of the tree.

2.2.2 *Structural Attributes and Data Attributes of a Tree*

The analytical (low-level) tasks in tree visualization are designed to acquire information about the structural and data attributes of a tree [24]. The structural attributes provide information about the "topology" of a tree and the data attributes provide information about the data associated with the nodes and links of the tree. For example, in the task "What is the height of the tree?", the user wants to inquire about a tree's structural attribute ("height"). In another task, "Find the child node with the maximum value." the user wants to identify the node with the highest "value" for the data attribute.

Structural Attributes are associated with different levels of the tree. For instance, the attribute "height" represents a tree-level attribute and the "value" describes a node-level attribute. In Table 1, we present the common structural properties. For additional information, we also define the properties ("Definition"), identify the part of the tree they are associated with ("Attribute Of"), list out the data type ("Data Type") and different aliases ("Also Known As").

Data Attributes are associated with nodes and links in a tree. For tree visualizations, categorical attributes are used to show textual information about the node and the link. Quantitative data records show numerical values associated with the nodes and links.

2.3 Common Tree Visualizations

The tree visualization design space is enormous [19], [25]. As a result of this large design space, there are many dimensions to classify tree visualization designs. Some common factors that are well known within the literature are the parent-child alignment technique (enclosure, indentation, adjacency), layout (linear, radial), dimensionality (2D v. 3D), and coordinate system (Cartesian, Hyperbolic) [1], [19], [25], [26]. A detailed classification of the tree visualization designs on all these factors is beyond the scope of the paper. Therefore, in this section, we explain common tree visualizations primarily grouped by their *parent-child encoding* technique. The parent-child encoding technique significantly changes the appearance of a tree visualization, making it the most common dimension for differentiating tree visualizations [1]. Within each parent-child encoding technique, we discuss other techniques that we encountered more frequently in the surveyed papers. A representative depiction of each technique is shown in Fig. 2.

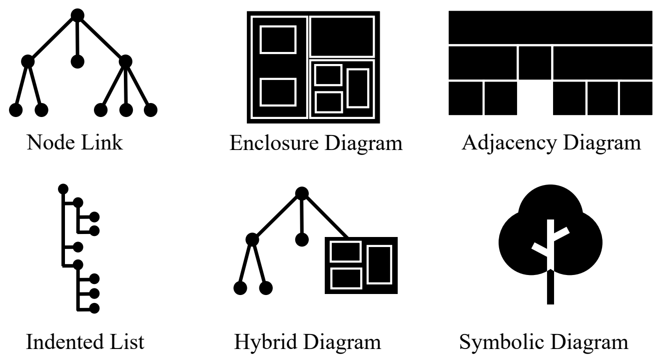


Fig. 2. Common tree visualizations grouped by parent-child encoding. More information regarding each encoding is available in Sec. 2.3. The sample "Hybrid Diagram" is inspired from the work of Zhao et al. [30].

Node-link Diagram: A node-link diagram represents nodes distributed in space with links encoded by connected lines. Space is often used to communicate hierarchical orientation, typically towards authority or generality. Usually, a 2D linear or radial space is utilized to break apart the breadth and depth of the tree. In some representations, 3D spaces may also be used to represent nodes of a tree visualization [27]. Another form of node-link visualizations is a hyperbolic diagrams [26]. They use non-euclidean geometrical surfaces for rendering the node-link visualization.

Enclosure Diagram: An enclosure diagram encodes the structure of data using spatial enclosure. An enclosure, which often uses rectangles (among other shapes) signifies hierarchy. Enclosure diagrams provide a single view of an entire tree, making it easier to spot large/small nodes. However, it is difficult to accurately read "depth" on an enclosure diagram. Techniques such as 3D treemaps (StepTree) exist to counter the effect of depth perception for treemaps [28].

Adjacency Diagram: An adjacency diagram makes use of adjacency and alignment to create a tree structure. An adjacency diagram uses the recursive subdivision of space. In this idiom, higher-level nodes get a larger area and child levels are constrained to the parent's extent. An adjacency diagram that uses vertical alignment of nodes is called an icicle plot [29]. The radial version of adjacency diagrams are commonly called sunburst or ring charts [4].

Indented List: An indented list diagram uses indentation to show parent-child relationships and it places all data items along vertically spaced rows. This type of visualization is commonly used as a component in an interface such as Windows File Explorer.

Hybrid Diagram: A hybrid tree diagram [30] interleaves two or more tree visualization encodings to exploit advantages of a particular method and mitigate disadvantages of the other.

Symbolic Diagram: A symbolic diagram uses realistic traits of biological trees to represent a tree dataset. These realistic trees represent entities as leaves or fruits and use trunks, branches, and offshoots to represent connections among different entities. As these trees are symbolic of real trees, we call them symbolic diagrams [19], [31].

2.4 Tree Visualization Task Space

Tree visualization tasks can be classified into 4 categories, corresponding to the combinations of {single, multiple} x {static, dynamic} trees. Each of these characteristics is defined as follows:

Single Static Tree: In single trees, users are interested in examining the structural(topology) and data(attribute) properties of a single tree [24]. In this category, the data is static, i.e. both structure and data values do not change with time.

Multiple Static Trees: In multiple trees, users want to compare two or more static tree visualizations [32], [33], [34].

Single Dynamic Tree: In single dynamic trees, a single tree evolves with time [35] and users are interested in analyzing the change of structure or data properties of a single tree [24].

Multiple Dynamic Trees: In multiple dynamic trees a user wants to compare two or more dynamic trees. These trees are an extension of the multiple trees and dynamic nature of the data. We did not encounter an example of this type in our review of the literature. Therefore, this category is a logical extrapolation, and tasks in this category can be concerned about tracking temporal changes in multiple hierarchies.

In this paper, we survey tasks for *single tree visualizations* because single trees are one of the most common tree visualization encodings [25] and they form the basic building block of tasks in all the other categories. For instance, to compare the topology of two trees, users first need to analyze the topology of each tree and then compare them to identify differences.

3 Survey of Literature for Tree Visualization Tasks

We contribute a collection of 212 tree visualization tasks based on a survey of published visualization research papers. As the amount of literature published about trees is vast, including across disciplines, we developed a literature search methodology targeted at identifying a sub-set of task-specific research articles for tree visualizations published in the visualization community. In this section, we summarize and justify our survey methodology and process, and present the results of a meta-analysis of the winnowed *54* publications included in our survey.

3.1 Literature Corpus Collection

Search Phrases: To enable a consistent search and use of the most commonly accepted vocabulary by the visualization community, we identified the appropriate keywords for tree visualization papers with tasks from the IEEE VIS Conference Reviewing system. The keywords used by IEEE VIS 2019 are standardized by its parent sponsoring society the IEEE Technical Committee on Visualization and Graphics (VGTC). We identified the terms "Tree Data", "tree/hierarchy structure", and "Task Abstractions" (without the quotes). To create semantically meaningful phrases from the keywords for identifying relevant papers, we joined the concepts together to create two search strings: "Tree Visualization Tasks", and "Hierarchy Visualization Tasks."

Search Space: Initially we used the Google Scholar search engine to identify relevant papers. However a search of the phrase "Tree Visualization Tasks" returns 324,000 results which, although highlighting the abundance of tree visualization research in the academic community, is an unrealistic number of papers to review. We instead focused our literature search by inputting our two search phrases into four specific relevant digital libraries: IEEE Xplore, ACM Digital Library, the Eurographics Digital Library, and the Wiley Online Library. A search of these two phrases in each of these four search engines yielded 555 papers. To ensure that we did not miss any relevant work or bias our search, we also reviewed the top 100 results based on relevance from Google Scholar. Finally, we also included all of the papers on Treevis.net [25] as it is one

Library/Source	Journals and Conferences	Papers
IEEE Xplore	IEEE Transactions on Visualization and Computer Graphics, Information Visualization (SAGE), IEEE Vis, Pacific Vis	91
Eurographics DL	Computer Graphics Forum, Eurovis and Co-located workshops	203
ACM DL	CHI and VINCI	201
Wiley DL	Software: Practice and Experience and Journal of Software Evolution Process	60
Google Scholar	-	200
Treevis.net	-	306
		Σ 1061

TABLE 2
Tally of tree visualization task papers identified from common digital libraries. We also included papers from Google Scholar and treevis.net to include articles that may not be available in the digital libraries.

of the most comprehensive tree visualization encodings surveys to date which covers over 300 tree visualization techniques.

In total, these resources yielded 1061 papers to review for our survey. Papers surveyed corresponding to the digital libraries and additional search sources are shown in Table 2. For each digital library, we explicitly state the list of journals and conferences where the surveyed papers appeared. For transparency, we discuss the search process and yield for each resource used in the Supplemental Material. The full corpus of 1061 collected papers was then winnowed for relevance to this task-focused survey through our exclusion and inclusion criteria.

3.2 Process and Rationale for Selection of Final Survey Corpus

We applied to the initial search space of 1061 papers two rounds of winnowing to identify papers that were "task-centric" with a focus on "singe-tree visualization". In this subsection, we discuss the *exclusion* criteria that allowed us to remove any paper that did not focus on tasks and the *inclusion* criteria that focused on collecting papers with tasks for "single-tree visualizations". In the next two subsections, we discuss the criteria developed and the methodology followed by the paper's co-authors.

3.2.1 Excluding Non-Task-centric Papers

In the first step of winnowing the corpus of tree visualization publications collected, we excluded papers that did not contain tree visualization task information.

Exclusion Criteria: Due to the use of the word "tasks" in our search phrase, many of our gathered publications were task-based but were not related to tree visualizations (e.g., [36], [37]). The search space also contained a large number of "technique/algorithm" papers. These technique papers focused on validating the tree drawing algorithm rather than the tasks the tree visualization support (e.g., [38], [39]). Several of these research articles did not claim tree visualization as a contribution, however, we found that these papers used tree visualizations to present their paper's results (e.g., [40]) or to explain a concept (e.g., [41]) in their domain. We removed papers from our survey in which the tasks did not relate to tree visualizations, the paper did not include a tree visualization task, or the tree visualization was a secondary contribution or discussion point for the paper.

Procedure: During this process, the authors used the exclusion criteria and excluded the papers that did not meet the criteria. The first three authors of this paper individually read the abstract, the keywords, and skimmed each of the 1061 papers to perform the exclusion process. In the end, the authors collectively discussed their final list and concluded that the **110** filtered articles should be analyzed in-depth for the task survey. The process took around three months. During this period, the authors were also working on other projects.

To assist collaboration among the authors, the papers were saved on a shared drive. Furthermore, we created a shared spreadsheet with a list of the 110 papers with metadata for each paper including the venue of publication and doi.

3.2.2 Including Single-Tree Visualization Papers

In the final winnowing stage, all the papers filtered as discussed in Sec. 3.2.1 were further scrutinized to identify the papers which contain static single-rooted trees.

Inclusion Criteria: We examined the data model for all 110 surveyed papers. If the data model strictly conformed to our scope, i.e., it was a single rooted tree hierarchy, then we identified the analytical tasks discussed in the paper. We found that publications generally fell into one of two common contexts of task specificity: The first category explicitly outlined the analytical tasks through the means of motivation or evaluation. We included all papers that elicited the tasks users can perform with a tree visualization. The second category of tree visualization papers that we included did not have an explicit list of tasks. For the second category of papers, we thoroughly analyzed the prose of the manuscript, and only if we were able to identify an analytical goal in the prose we included the paper in our survey. We discuss the exact process of task collection and the papers selected using the discussed technique in detail in Sec. 3.4.

Procedure: In this stage, the first three authors analyzed the entire corpus of all 110 papers. During the review, the authors read the paper and marked each paper's data model (i.e., single tree or any other form of tree or network) and whether the paper includes tree-specific user tasks. After individual analysis of the manuscripts, the first three authors collectively discussed the final list and reached the final paper corpus containing a total of **54 papers** (see Table 3). In the Supplemental Material, corresponding to each of the 110 papers, we provide the reason for exclusion. This process took around two months.

3.3 Meta-Analysis of Surveyed Papers

For each of the 54 papers in our survey, we recorded relevant publication information as well as information about the tree visualizations discussed in the paper. In this section, we present a summary and meta-analysis of the data collected.

Meta-data Collected: The publication information recorded for each paper includes their year of publication, author names, abstract, DOI, and the type of paper. The tree content information contained in and recorded for each paper includes evaluation type and the visual encoding used in the paper. A list of the surveyed papers with associated metadata specifically related to the task information are listed in Table 3. For the complete collection of metadata, please refer to the Supplemental Material and https://osf.io/u5ehs/

Temporal Analysis: Our survey contains papers published from 1992 to 2020. We observed a steady publication trend with no significant variation. In the period 2000-2009, we observe a

TABLE 3
The literature corpus of 54 papers used for the task survey. The columns "Paper Type" and "Evaluation Type" identify the primary contribution and the evaluation method of the paper, respectively (Sec. 3.3). The visualization techniques used in each paper are marked with "X" (Sec. 2.3). "Task Collection" lists the method of task collection employed for the paper (Sec. 3.4).

Paper	Paper Type	Evaluation Type							Task Collection
Brian Scott Johnson [42]	Empirical	Mixed		X		X			Explicit
Barlow & Neville [43]	Empirical	Quantitative	X	X	X				Explicit
Alfred Kosba [44]	Empirical	Mixed	X	X	X			X	Explicit
Stasko et al. [4]	Empirical	Mixed		X	X				Explicit
Burch et al. [45]	Empirical	Mixed	X						Explicit
Zhao et al. [30]	Technique	-					X		Non-Explicit
Plaisant et al. [46]	Technique	Mixed	X			X			Explicit
Cockburn & McKenzie [27]	Empirical	Mixed	X			X			Explicit
Wang et al. [47]	Empirical	Insight	X	X					Explicit
Muller et al. [48]	Empirical	Eye-Tracking	X	X	X				Explicit
Andrews & Kasanicka [49]	Empirical	Mixed	X	X		X			Explicit
Ham & Wijk [50]	Technique	Mixed		X	X				Explicit
Ziemkiewicz & Kosara [51]	Empirical	Quantitative	X	X					Explicit
Bladh et al. [28]	Empirical	Quantitative		X	X				Explicit
Holten [52]	Technique	Insight	X	X					Non-Explicit
Novick [53]	Empirical	Eye-Tracking	X						Non-Explicit
Wang & Parsia [54]	Application	Quantitative	X	X					Explicit
Gortler et al. [55]	Technique	Case-Study		X					Non-Explicit
Forbes & Dang [56]	Technique	Case-Study	X						Explicit
Blanch et al. [57]	Technique	Insight	X						Non-Explicit
Auber et al. [58]	Application	Qualitative	X	X					Explicit
Block et al. [59]	Application	Insight	X						Explicit
Linsen & Behrendt [60]	Technique	Quantitative					X		Explicit
Tuttle et al. [61]	Technique	Qualitative	X		X				Explicit
Song et al. [62]	Empirical	Mixed	X				X		Explicit
Gomez et al. [63]	Application	Case-Study			X				Non-Explicit
Robert Theron [64]	Application	Case-Study	X						Explicit
Tan et al. [65]	Application	Mixed	X						Explicit
Neumann et al. [66]	Technique	-		X					Non-Explicit
Wiss et al. [67]	Empirical	-	X	X	X				Explicit
Soares et al. [68]	Technique	-		X					Non-Explicit
Chen et al. [69]	Application	Mixed			X				Explicit
Elzen & Wijk [70]	Application	Case-Study	X						Non-Explicit
Wattenberg & Viegas [71]	Application	-	X						Non-Explicit
Burch et al. [72]	Technique	Mixed	X		X				Explicit
Sallaberry et al. [73]	Application	Case-Study						X	Explicit
Woodburn et al. [74]	Empirical	Mixed		X	X				Explicit
Beheshti et al. [75]	Empirical	Mixed	X			X			Explicit
Santana et al. [76]	Empirical	Eye-Tracking		X					Non-Explicit
Bladh et al. [77]	Empirical	Mixed		X					Explicit
Wiss & Carr [78]	Empirical	Mixed	X	X					Explicit
Santos et al. [79]	Empirical	Mixed	X						Explicit
Biuk-Aghai et al. [80]	Empirical	Mixed		X					Explicit
Liang et al. [81]	Technique	Quantitative		X					Explicit
Wetering et al. [82]	Technique	Mixed			X				Explicit
Li-Wei et al. [83]	Technique	Quantitative			X				Explicit
Shin et al. [84]	Technique	Mixed	X			X			Explicit
Band & White [85]	Application	Mixed				X			Explicit
Andrews et al. [86]	Application	Quantitative		X		X			Explicit
Dang et al. [87]	Empirical	Mixed	X	X	X			X	Explicit
Muramalla et al. [88]	Empirical	Mixed			X				Explicit
Golemati et al. [89]	Empirical	Quantitative				X			Explicit
Long et al. [90]	Empirical	Mixed		X					Explicit
Heinicke et al. [91]	Empirical	Mixed	X	X	X				Explicit

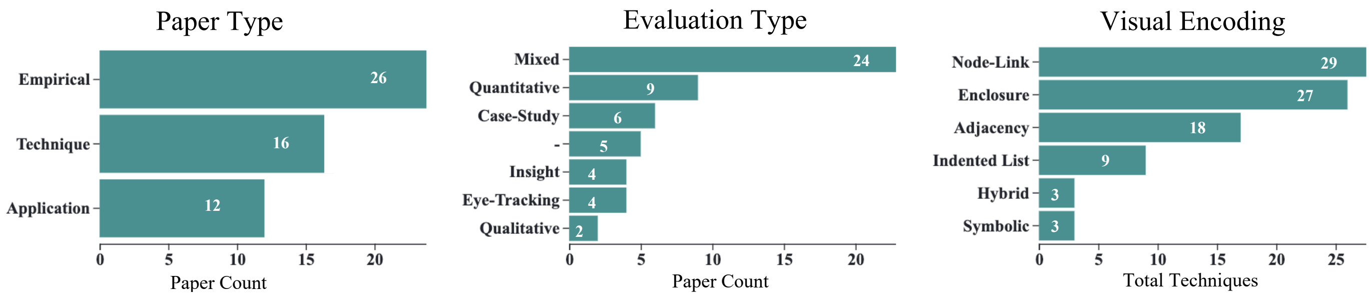


Fig. 3. In this figure we present an analysis of surveyed papers dataset (Sec. 3.3). The left-most Paper Type plot shows task data is spread across a wide-range of paper types. The Evaluation Type plot shows the specific types of evaluations included in the surveyed papers. In the evaluation type plot, papers without evaluation are marked with "-". The right-most Visual Encoding plot summarizes the visual encodings evaluated with analytical tasks.

trend in evaluations of tree visualization tasks with common tree visualization encodings including node-link diagrams, treemaps, icicle plots, indented list, and sunburst charts [4], [27], [43], [44], [49]. From 2010 onwards, the trends have evolved and evaluations have moved beyond basic tree visualizations.

Paper Type: Categorization of papers by common "type" provides a high-level overview of the content of the research paper. As paper type is not a part of standard meta-data provided by the publication venues, in our survey based on the abstract and keywords of each paper we assigned a type to each paper. To choose the paper type we used the list of paper type options in the Call For Participation (CFP) of the IEEE VIS 2019 Conference. As shown in Fig. 3, the most common type of paper was "Empirical Study" (also known as Evaluation) papers (26) which compared two or more tree visualization encodings on a set of analytical tasks. The next most common type of paper in our survey was "Technique paper" with a human-centered evaluation (16). Technique papers were closely followed by "Application & Design Study" papers (12) which included tasks that motivated their design. Our survey does not contain "System" paper type, possibly due to our inclusion criteria that focused on the selection of papers with single static trees, and system papers are usually large complex software with multiple visualizations interacting with each other.

Evaluation Type: In our survey, as shown in Fig. 3, 49 papers contained an evaluation study (e.g., quantitative evaluation, qualitative evaluation, case-studies). Empirical evaluations can provide quantitative data (9 papers), or a mixture of both quantitative and qualitative data (24 papers). In addition to empirical evaluations (35 papers), we found three other types of evaluations: insight (4 papers), eye-tracking (4 papers), and case-studies (6 papers). An insight-based evaluation measures the number of analytical insights, like patterns, trends, similarity, etc., related to data or structural properties of a tree visualization. Eye-tracking studies use eye-movement as a measure to understand cognitive strategies applied by users to solve a typical hierarchy exploration task. The final type of evaluation in our survey is case studies. Case studies demonstrate the application of a tree visualization in a domain specific hierarchical data exploration task. Mixed evaluations, that contained both quantitative and qualitative measures, were the most common type of evaluation study. Availability of qualitative measures allow visualization researchers to understand the preference for visualization techniques, which can be important specially when the quantitative measures do not vary significantly. Moreover, we noticed that eye-tracking studies give a holistic picture of the cognitive strategies employed with tree visualizations. However, existing eye tracking studies have only evaluated a limited number of encodings including node-link [48], [53], enclosure (treemap) [76], and adjacency (icicle plot) [48]. Therefore, more effort is required in the space of eye-tracking studies for tree visualizations, and studies should include other common tree visualization encodings like indented lists and radial adjacency diagrams (sunburst chart).

Visual Encoding: For each surveyed paper we recorded the type of visual encodings included in the publication. By recording this information, our survey can support in future work the construction of a high-level mapping of how well different tree visualization techniques support various tasks. Fig. 3 (Visual Encoding) breaks down the papers according to the visual encoding used.

3.4 Task Collection

Based on the papers in our literature survey (Sec. 3.3), we contribute a curated dataset of tree visualization tasks. For each paper in our survey, we extracted the tree tasks included in the paper from contexts including motivation for new techniques, taxonomy of tasks in new domains, and evaluation studies of tree visualizations. This novel tree task dataset enables the identification of general tree visualization tasks and serves as an effective resource for tree design and evaluation. An analysis of the task dataset also reveals well-studied areas of tree visualizations as well as areas for future research (Sec. 5). For the extraction of tasks from our surveyed papers, we grouped the tasks into two categories: *explicit tasks* and *non-explicit tasks* based on the format of task availability in the paper, see Table 3 (Task Collection). All of these tasks, and whether they were collected as an explicit or non-explicit task, are available in our dataset provided in the Supplemental Material and on https://osf.io/u5ehs/

Explicit Tasks: In our survey, 43 papers contain an explicit list of analytical tasks in the body of the paper. Explicit lists are most commonly included in evaluation studies, or design studies with an evaluation component, and are often used to evaluate the performance of a visualization in terms of task accuracy and completion times. For example, Stasko et al. [4] present analytical tasks phrased in naturally accessible language such as, "Find largest files or directory", "Find the deepest directory", and "Compare files/dirs by size" as part of an evaluation study.

Non-Explicit Tasks: In our survey, 11 papers did not contain an explicit list of analytic tasks in the paper. The tasks in these papers are typically mentioned in the prose of the paper, such as an introductory or case-study explanation section of the paper, and required a trained visualization researcher to identify the tasks. For our survey, the paper's first author read each of these papers in order to carefully extract the relevant text for the tree tasks. In

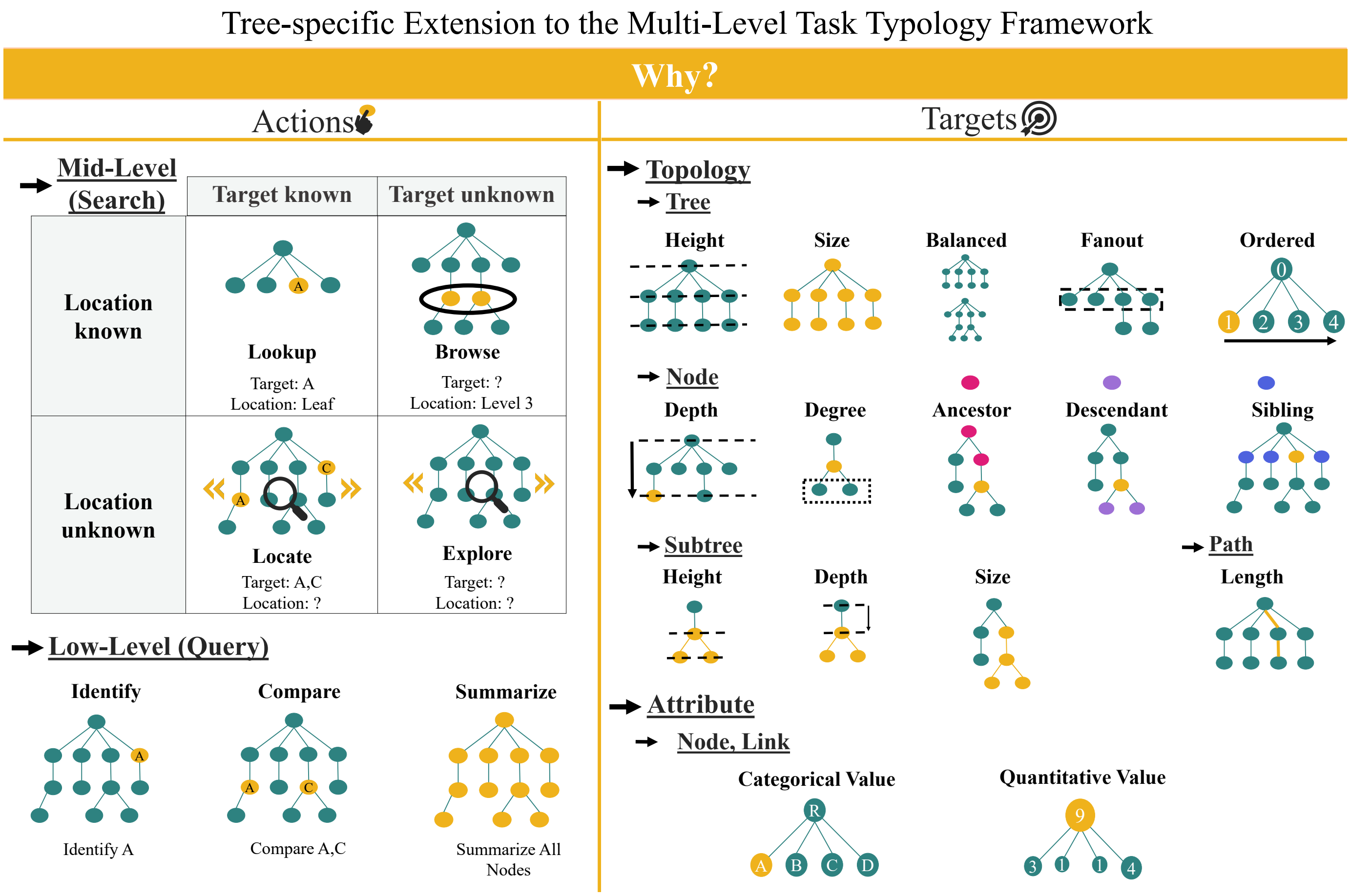


Fig. 4. Summary of tree visualization tasks broken down by *Actions* and *Targets*. The Actions use the Multi-Level Task Typology (MLTT) [1], [2] terminology to identify the types of actions users can perform in tree visualization tasks. The Targets include a novel *Nested-extension* of the existing MLTT target characterization that adds specificity for tree visualizations.

our task dataset we include a closely paraphrased summary of the task. The paraphrased summary is an objective representation of long task verbatim that we collected from the paper. However, to ensure transparency, we also report the exact verbatim in the task dataset. For example, Bubble Treemap [55] presents an analysis of the S&P 500 index stocks. In the case study, the domain analytical task was described in the prose as: "By looking at the waviness of the contours, it is relatively easy to identify the stock with the biggest changes, since the variance is reflected in all the contours of the respective sub-systems." Based on our understanding of the main contribution of the paper and careful analysis of discussed applications, we presented the task in our survey as "Identify distribution of the node value".

4 Task Abstraction Process & Theory

In this section we discuss the Multi-Level Task Typology (MLTT) framework (Sec. 2.1) in the context of tree visualizations and present our novel extension to the framework. This tree-specific extension of the MLTT framework is visually summarized in Fig. 4. The new extension contributes the required target specificity for the effective description of tree visualization tasks.

4.1 Tree Visualization User Goals as Actions

The papers in the survey (Sec. 3) varied widely in terms of application domain ranging from Biology to Computer Science to Social Science. We translated domain specific tasks to a consistent abstract terminology. For example, the tasks:

- Which directory has more files directly inside: "/hcil/spotfire" or "/hcil/spacetree"? [49]
- Is "Interest on the Public Debt" more than half of the "Department of Treasury"? [42]

both translate to the same abstraction: "Lookup the nodes to compare degree of the topology". This transformation of tasks from domain-specific to abstract language is essential to enable visualization creators to effectively compare tasks across different domains, look for relevant techniques and strategies in different application areas, and analyze the general trends around human-centered tasks for tree visualizations. We adopted the MLTT framework [1], [2] to accomplish the analysis and abstraction of the tasks. We now discuss each part of the "Action" framework in MLTT in relation to tree visualizations:

1. **Analyze (High-level Action)**: In our survey we did not identify any "produce" tasks. All of the tasks in our survey are instances of "consume" in which the visualization communicates information already known or derived. Within consume, the tasks in our survey were being used to test or generate a hypothesis so they have the high-level goal of *discover*. A notable observation is that in some cases, the high-level task classification can be challenging to identify as the high level task is ambiguous. This ambiguity arises due to insufficient discussion in the paper about the visualization creator's

goals for a particular visualization. However, we argue that the objective of the task remains both unhindered and heavily dependent on the mid- and low-level actions rather than the high-level actions. This is in-line with what Munzner states about the concept of the target being explicit with search (mid-level) and query (low-level) actions but more implicit with what the user presents or discovers [1].

2. **Search (Mid-level Action)**: The mid-level action indicates the type of search a user must perform in order to find the target. What type of search is determined by whether the target is known, and whether the location of the target is known in the task description. The mid-level targets are illustrated in Fig. 4 (Actions). To perform the classification of search, we define the concept of "target known" and "location known" for trees:

Target Known: Target is known when the task explicitly mentions the identity of the target. For tree visualizations, the targets are "Tree", "Node" and "Subtree" (Sec. 2.2). For single trees, tree-level targets' identity is always known because there is just one tree to analyze. For example, in the task "Users had to indicate the total number of levels in the tree" [50], it is evident the target is the entire tree visualization. The node identity is known if the task explicitly mentions the label or a unique identifier of the node. For example, in "Find node entitled 'cannibalism'" [85] the node label is available in the task description. An example of implied information is "Identify the largest node" [4] which reveals that the node represents data with the maximum value and is therefore unique and distinguishable. A subtree target is known if the task provides information about the root node of the subtree. For example, in "Which bottom-level directory of the 'teaching' subtree has the largest number of files?" [60] the root node of the subtree (teaching) is explicitly mentioned in the task.

Location Known: The location is known if the task explicitly mentions the position of the target within the tree. For tree level tasks, the location is always known because all nodes and links have to be considered. For node and subtree level tasks, the location is their distance from the root node. For example, "What is the name of the largest level 4 item in the entire '1992 US Budget?'" [42]. Another way to identify the location of the node is through the path. For example, "Find the directory 'yidemo' (/hcil/lifelines/yidemo)?" [49] outlines the path to the node from the root of the tree. Tasks related to leaf nodes implicitly disclose the location of the node. For instance, "Users had to select the three largest leaf nodes" [50]. In this task, users know that leaf nodes are located at the bottom of each branch.

Based on the target and location information, we discuss examples of each each of the four mid-level search scenarios of *lookup, locate, browse,* and *explore*:

Lookup: When the task provides information about the target and its location we classify the task as a *lookup* task. *Example:* "What is the value of 'Health Care Financing Administration' (level 4) under 'Department of Health'? (level 3)?" [42] The node location (level 4) and exact label of the target node ("Health Care Financing Administration") are provided in the task description.

Locate: When the location is not specified in the task but information about the target is provided, it is classified as locate. *Example:* "Identify the largest (size) node." [4] In this task the largest node gives information about the target, but the location is not specified.

Browse: When the target is not specified in the task, it is classified as browse. *Example:* "Name the seven level-4 items exceeding 50,000 in order of decreasing value." [42] This task gives information about the location of the target, but users still must find the target.

Explore: When the target and the location both are unknown, the task is classified as explore. *Example:* "Find the number of directories WITHOUT a file of type .js." [44] In this task, users do not know the target or its location.

3. **Query (Low-Level Action)**: After a user finds the target(s) by performing the mid-level search action, the user's proceeding aim or low-level goal, see Fig. 4 (Action), is to investigate the target(s) by either identifying, comparing, or summarizing:

Identify: When the task is related to the identification of a single target, it is classified as *identify*. *Example:* "What is the name of the largest level-4 item under 'Department of Defense Military'?" [42]. The mid-level goal is to lookup all the items of level 4 (known target) under the node "Department of Defense Military" (known location). Finally, the low-level action is to identify the value of the largest item in the node.

Compare: When the task is related to the comparison of two or more targets, it is classified as compare. In this action, targets must be specified. *Example:* "At level 2 which is larger: 'Cabinet Agencies' or 'Miscellaneous Agencies'?" [42]. The mid-level action is to lookup the nodes (targets known) "Cabinet Agencies" and "Miscellaneous Agencies" in level 2 (location known). Comparing these two nodes to determine which is larger is the low-level goal of the user for this particular task.

Summarize: When the task is related to the full set of targets then it is classified as summarize. *Example:* "Binary or n-ary tree?" [43]. The mid-level goal is to lookup because the user is analyzing all the nodes of the tree (target and location known). The low-level action is to summarize because the task requires a determine the degree of all the nodes of the tree.

4.2 Tree Visualization Targets

According to the MLTT framework, tree visualization targets can be broadly classified into two categories: *topology* and *attribute* [1], [2]. Topological targets relate to the tree structure, and attribute targets relate to the data values. However, we argue that the classification of targets as topology or attribute is not sufficiently descriptive to precisely describe the tree visualization task.

To illustrate this point, we present the task abstraction of two different analytical tasks with the MLTT framework in Fig. 5 (A). Lets consider Layla who wants to answer the following two questions about her organization's structure: "Are the departments balanced?" and "How many employees are in the CTO group (i.e., a specific branch)?" Layla then uses the MLTT to classify these two tasks, and finds that the targets in both tasks map to *topology*. Based on the abstraction, Layla is confused about which representation to pick. She could in theory choose a list diagram, an adjacency icicle diagram, an enclosure diagram, or a node-link diagram to show the tree topology. In this case the target "topology" is too general to help Layla more precisely identify her target and subsequently the optimal visual encoding choice. Using our new extension to MLTT for target specificity for tree visualization tasks, as shown in Fig. 5 (B), Layla is able to see that her first question abstracts to a task about the balance of the organizational chart thus a node-link diagram is a most appropriate choice. Her second task relates instead to the size of subtrees, a completely different target, thus an indented list is an appropriate visual encoding choice to complete her task.

To increase the specificity for tree targets, we add two levels of sub-classification to the original topology and attribute-based targets for MLTT. Next, we present a nested multi-tier abstraction technique for tree targets:

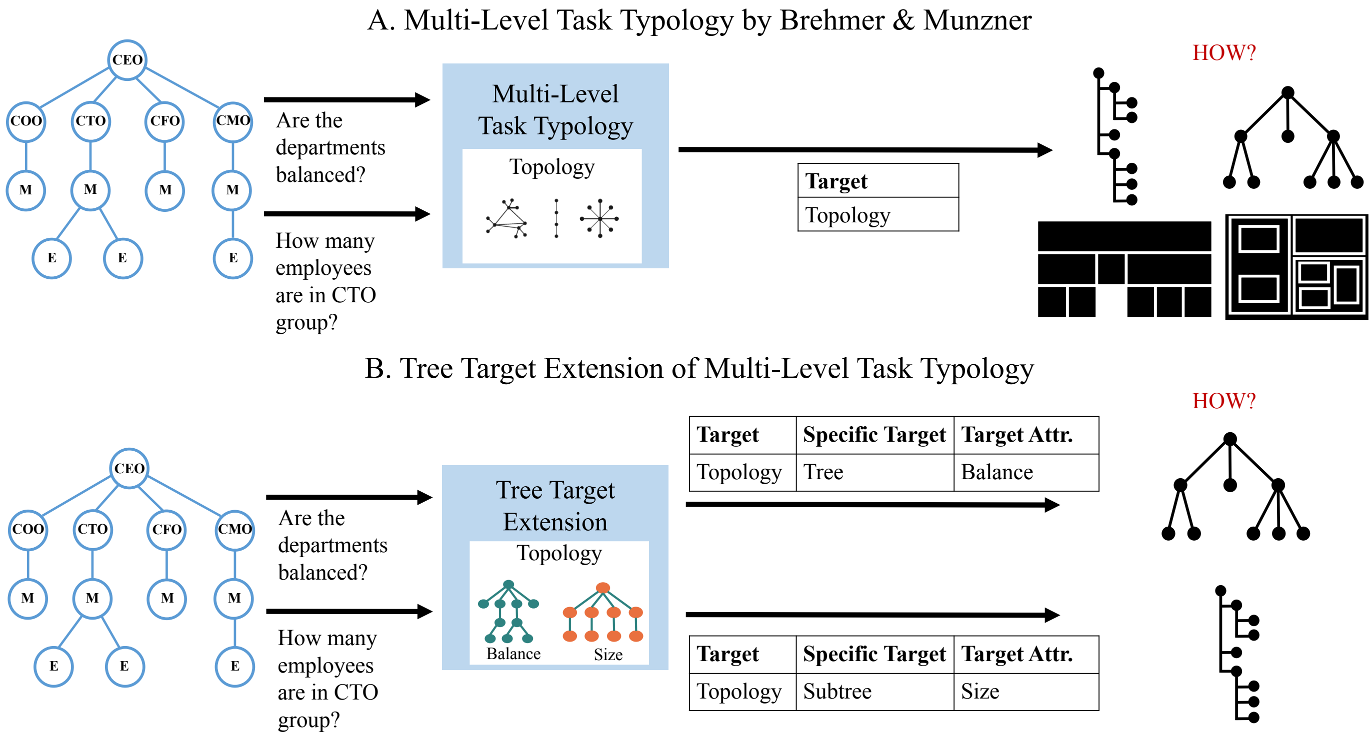


Fig. 5. The figure (A) shows how Multi-Level Task Typology abstracts two tasks which are related to different topological attributes(balance and size) to a blanket topology category. High-level target of topology limits our understanding of target of interest. Consequently, the abstract definition does not provide enough information to choose the correct visual encoding. To add more target specificity we extend topology definition of Multi-Level Task Typology. In (B), we add tree visualization specific targets to the taxonomy. Target expansion adds more specificity to the task to better support evaluation and design.

1. Target (High-Level): The MLTT framework, as well as the tree-specific General Tasks Applicable to Most Trees framework by Fekete & Plaisant [24], use "topology" and "attributes" as the basis of categorization for tree visualization tasks. Therefore, our extension maintains the existing MLTT high-level target classification.

2. Specific Target (Mid-Level): In the mid-level, a more specific target type is assigned to the high-level target. In our extension, this mid-level specificity identifies the precise part of the tree. For example, a topology task to find the degree of a node is broken down such that researchers and practitioners can explicitly identify that the main item of interest is a node. The original MLTT framework does not include a list of tree-specific targets in their original list of targets, nor does it discuss the idea that targets can be multi-level. Some existing taxonomies for graph visualization tasks [7] and tree tasks [24] divide the topology and attribute targets based on specific objects. For example, Lee et al. divide the graph targets into graph-specific objects including nodes, paths, and connected-components [7]. This decomposition of the topology and attribute targets into specific graph-based objects provides more clarity about the nature of task. Our mid-level *specific target* similarly breaks-down tree visualization targets. The mid-level tree-specific targets in our extension were created through a grounded theory approach [92] applied to the tasks in our curated dataset. For each task in our dataset (Sec. 3.4), we identified the target of the task. For example, in the task "Is the tree balanced or unbalanced?" [60] the target is the entire tree. In the task "Which directory includes a deeper hierarchy: Flute or Guitar?" [44] the targets are the flute and guitar subtrees. Through this process we identified the specific targets for tree topology tasks: *tree*, *subtree*, *node*, and *path* (i.e., sequence of nodes and links). The specific targets for tree attribute tasks are: *node* and *link*. These mid-level specific targets will be discussed in further detail in Sec. 4.3.

3. Target Attribute (Low-Level): At the lowest-level, additional specificity is given to describe the *specific target*. Each mid-level specific target can have multiple characteristics. For example, a node can have topology characteristics (e.g., degree, ancestor, siblings) or data characteristics (e.g., quantitative value, qualitative value). Our novel extension of MLTT for trees thus includes this explicit identification of the low-level target attributes. The target attribute is an essential part of the visual identification process of relevant feature to complete a task. With clear identification of the target attribute, a visualization creator can choose a visual encoding method that enhances the target attribute to assist with task completion.

At the start of this Sec. 4.2, we illustrated the need for greater target specificity to accurately describe a task. We now return to this example to demonstrate how our new extension to MLTT for tree visualization can precisely and effectively describe tasks. As shown in Fig. 5, Layla wants to better understand her company's organizational structure. With our novel tree extension to the MLTT framework, we can now precisely describe Layla's tasks in consistent abstract task language (Fig. 5 B). Although both questions relate to the tree topology, the first task "Are the departments balanced?" has the specific target *tree* whereas the second task "How many employees are in CTO group?" has the specific target *subtree*. The target attribute adds further essential information to clarify that the first task ("Are the departments balanced?") requires a *lookup* action of the target attribute *balance* in the tree structure. Balance comparison can be supported by a tree visualization encoding that can be aligned by a central axis so that the height/width of all subtrees can be compared [93]. For instance, the Reingold & Tilford [94] node-link diagram factors in the comparison between the left and right sides of the tree and provides a symmetric tree visualization. This distinction of specific target and its attribute provides strong evidence to support the final encoding choice for the visualization.

In the second task ("How many employees are in CTO group?"), the task requires counting the total number of elements in a *subtree* with a target attribute of *size* (Sec. 2.2). Empirical evidence in the literature suggests that with indented lists participants are faster [49] and more accurate [44] when completing "size" lookup tasks. As with the first task, the additional information afforded by the specific target and target attribute can inform an effective choice of visualization encoding. As illustrated in this example, our novel multi-level target categorization not only supports task understanding but also facilitates the choice of visual encoding.

4.3 Specific Targets and Target Attributes

In our novel MLTT extension for trees, we contribute the concept of specific target and target attributes to provide the necessary task specificity for trees. Specific targets for tree visualization tasks are focused elements within a tree structure that more specifically identify the item of interest in a tree visualization task. Based on the literature survey, we found a range of specific targets for topology- and attribute-tasks (Fig. 4). In this section we present an exhaustive list of specific targets, and our rationale for choosing the specific target as a part of the extended framework. Furthermore, each specific target has certain additional attributes which support effective querying of a tree visualization to complete a task. For each specific target, we provide a list of these target attributes. To provide context, we also provide real examples from our task dataset. For a high-level overview of the target categories and the corresponding tree visualization tasks, please refer to Table 4.

4.3.1 Topology Tasks

Specific Target - TREE: In a topology task, when the level of focus is the entire tree, then the specific target for the task is categorized as *Tree*. Ultimately in a task participants are interested in extracting a structural property associated with the tree, as illustrated in the examples below. In the context of our framework, the structural attributes of the tree are defined as target attributes. **Target Attributes: Height, Size, Balance, Fanout,** and **Ordered**.

- "What is the maximum depth of the eBay hierarchy?" [44] (Specific Target: *Tree*, Target Attribute: *Height*)
- "Is the tree balanced or unbalanced?" [60] (Specific Target: *Tree*, Target Attribute: *Balance*)
- "Count all leaf nodes of the hierarchy." [48] (Specific Target: *Tree*, Target Attribute: *Size*)

Specific Target - NODE: In tree visualization topology analysis, some tasks have a focused scope, i.e., they are related to a specific node in the tree visualization. The structural attributes of nodes are discussed in Sec. 2.2. **Target Attributes: Depth, Degree, Ancestors, Descendants,** and **Siblings**.

- "Users had to indicate level of a predetermined node." [50] (Specific Target: *Node*, Target Attribute: *Depth*)
- "Which state has more entries on the topic of Transportation: Bavaria or Berlin?" [86] (Specific Target: *Node*, Target Attribute: *Degree*)
- "Identify siblings of a given individual." [79] (Specific Target: *Node*, Target Attribute: *Sibling*)

Specific Target - SUBTREE: A tree is made up of multiple sub-hierarchies called subtrees. Subtrees exhibit all the structural properties of a tree. In addition to the basic properties of tree targets, subtrees can also possess properties that are meaningful for nodes like depth and ancestors of a subtree. As a result of this unique properties of subtrees, our extension considers subtree as a specific target for topology tasks.
Target Attributes: Height, Size, Balance, Fanout, Ordered, Ancestors, and **Depth**.

- "Which directory includes a deeper hierarchy: 'Flutes' or 'Guitars'?" [44] (Specific Target: *Subtree*, Target Attribute: *Height*)
- "Determine which of two given directories contain the most file including subdirectories?" [28] (Specific Target: *Subtree*, Target Attribute: *Size*)
- "Find the deepest subdirectory inside the directory 'pad++' (/hcil/pad++)." [49] (Specific Target: *Subtree*, Target Attribute: *Depth*)

Specific Target - PATH A path in tree visualization is a sequence of nodes and links that shows the connection between a parent and child. We found that many tasks that appeared as path tasks were actually related to finding a node in the tree visualization. We discuss this phenomenon in more detail in Sec. 4.5. For *path* we identified only one target attribute. **Target Attribute: Length**

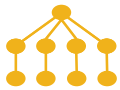
- "How many steps would you have to make to get between node labor affairs organizations and field hospitals?" [85] (Specific Target: *Path*, Target Attribute: *Length*)

4.3.2 Attribute Tasks

In many tree visualization tasks, the main focus of interest is a data attribute that is a part of the original hierarchical dataset. For instance, in an organizational chart, the data attribute could be salary, age, and education qualification for each employee. We classify all data value identification tasks as "attribute" tasks. In the task survey data, a large proportion of tasks (48.5%) had a quantitative or qualitative value represented on the nodes in the tree visualization. For instance, in the task "Find the collection 'Elections' in north Rhine-Westphalia" [86], the users had to find a node named "Elections". Therefore, we realized that "attribute" tasks also need a mid-level sub-division to add specificity to the keyword "attribute". Based on the common task descriptions we divide the high-level attribute target into two mid-level specific targets: **Nodes** and **Links** (Fig. 4).

Specific Target - NODE: Visualization creators can encode attributes in hierarchical datasets on a node in tree visualization. In tree visualizations, nodes are more frequently used for encoding data attributes. The usage of a node for mapping data attributes is expected because generally tree visualization encodings have a visual mark to represent the nodes, see Sec. 2.3. The nodes represent data attributes by visually encoding the values as different channels like the size, area, or color. **Target Attributes: Quantitative Value, and Categorical Value**.

- "Find a node having the maximum attribute value of the second layer." [83] (Specific Target: *Node*, Target Attribute: *Quantitative value*)
- "Find the collection 'Elections' in north Rhine-Westphalia" [86] (Specific Target: *Node*, Target Attribute: *Categorical value*)

Target	Specific Target	Target Attribute	Example Tasks
Topology	Tree 	Balance	Identify if the level of any two leaf nodes differ by at most 1. LK ID
		Height	Find the height of the tree. LK ID
		Size	Count the total number of nodes in the tree. LK ID
		Fanout	Identify the maximum number of nodes at any level of the tree. LK ID
		Order	Identify if all nodes are assigned a fixed ordering. For e.g. ordered by a categorical or quantitative value. LK ID
	Node 	Degree	Count the total child nodes of node 'A' at level 'Y'. LK ID
			Compare the count of total child nodes of node 'A' and 'B'. LC CO
			Explore and summarize if the tree is binary or n-ary. EX SU
		Ancestors	Identify the parent/s of node 'A' at level 'Y'. LK ID
			Identify the parent of node 'A' in the tree. LC ID
		Descendants	Identify the children of node 'A' at level 'Y'. LK ID
			Identify the children of node 'A' in the tree. LC ID
		Depth	Find the depth or level of node 'A' in the tree. LC ID
			Compare the depth of node 'A' and 'B'. LC CO
		Siblings	Identify the siblings of node 'A' at level 'Y'. LK ID
			Identify the node/s which have minimum N siblings at level 'Y'. BR ID
	Subtree 	Size	Compare the size of subtree rooted at node 'A' and 'B'. LC CO
			Identify the node/s that have minimum N descendants at level 'Y'. BR ID
		Height	Identify the height of the subtree with root node 'A'. LC ID
			Compare the height of subtree rooted at node 'A' and 'B'. LC CO
		Depth	Compare the depth of subtree rooted at node 'A' and 'B'. LC CO
	Path 	Length	Identify the distance between nodes 'A' and 'B'. LK ID
Attribute	Node 	Categorical Value	Identify the node with label 'A'. LK ID
		Quantitative Value	Identify the nodes with a value higher than X. BR ID
			Compare the value of node 'A' and 'B'. LC CO
	Link 	Categorical Value	Identify the link label 'L' connecting nodes 'A' and 'B'. LK ID
		Quantitative Value	Identify the value of link 'L' connecting nodes 'A' and 'B'. LK ID

Mid-Level Actions

LK Lookup LC Locate BR Browse EX Explore

Low-Level Actions

ID Identify CO Compare SU Summarize

TABLE 4

This table presents a high-level overview of the tree visualization targets that we discuss in detail in Sec. 4.3. As per the extended MLTT typology, the targets are broken down further into specfic target and target attributes. In this table we also list example tree visualization tasks corresponding to each target attribute. Each task example is appended with the corresponding mid- and low-action abstraction. The abstraction demonstrates that based on the mid- and low-level actions several variations of tasks can be produced for the target attributes. A full version of this table, with tasks corresponding to each combination of target and mid-level action is available in the Supplemental Material.

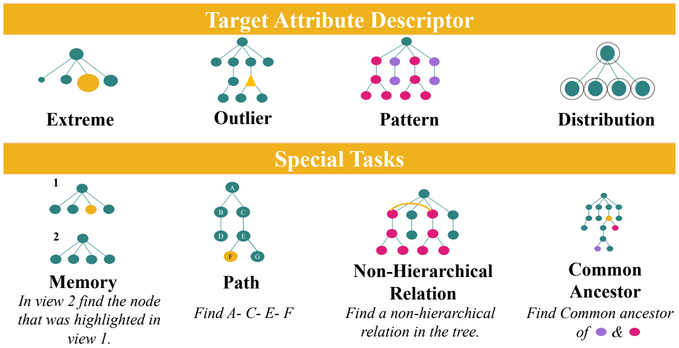


Fig. 6. Examples of *Target Attribute Descriptors* (Sec. 4.4) and *Special Tasks* (Sec. 4.5). For each special task, we also present a sample task description.

Specific Target - LINK: Link or Edges are the connection which represent parent-child relationships in a tree visualization. Links are only available for explicit tree visualization encodings (node-link, indented list, hybrid, and symbolic representations). In a tree visualization, links can represent all the data values a node represents. However, we found that links are less frequently used for encoding data attributes for tree visualizations. Consequently, in our literature survey we found no tasks related to links in a tree visualization. As we discuss further in Sec. 6, links can be a method to encode data and this lack of task examples represents an under-explored area in need of further research to evaluate the effectiveness of using links for encoding data. **Target Attributes: Quantitative Value, and Categorical Value**.

4.4 Target Attribute Descriptor

In our literature survey we found that some tasks describe additional characteristics for target attributes such as "What is the name of the **largest** Department (level 3) under Cabinet Agencies?" [42]. In this task, in addition to querying the quantitative value of each node at level 3, users also need to find the node with the largest value. In the MLTT framework [1], these analytical descriptors are assigned to data attributes. However, we found that the analytical descriptors can be valid for both structural and data attributes of a tree visualization. For example, "Find the **bushiest** child node." [54] is a topology task where the users need to find a node with the maximum degree. We argue that a clear distinction of target descriptor information in the task abstraction not only precisely defines the tasks but also assists to identify the best visual encoding for the task. For instance, Santana et al. [76] showed that an inverted scale treemap, where the size of each node is inversely proportional to its value, is a more effective visual encoding when the common goal of the user is to identify the maximum/minimum values. Therefore, if researchers know that the main goal of a visualization task is to identify maximum values, then they can use a treemap representation with an inverted scale. We identified four target attribute descriptors in our task dataset which are illustrated in **Fig. 6** and explained below:

Extreme: The extreme descriptor indicates that the task was related to finding the maximum/minimum value of a structural or data attribute. *Example*: "Find a node having the maximum attribute value of the second layer." [83]

Outlier: The outlier descriptor indicates that the task was about finding a node with an attribute different from a set of nodes. *Example*: "Outliers in terms of quantity of contacts (size of the small branches)" [73].

Pattern: A pattern descriptor represents a high-level trend that characterizes the data. For instance, a pattern for temporal data attributes can be a series of increases and decreases in the attribute value over time. In the context of tree visualization, patterns can be understood as a similarity, trend or order in the structure of the tree or the data attribute associated with nodes. For instance, an ordered tree has a topological pattern, where each leaf node in the tree is positioned based on the quantitative or ordinal value encoded on it. *Example*: "Identify patterns in node position at each level of the tree".

Distribution: Data attributes can store a distribution attribute instead of individual value. For instance, in an organization chart, an employee node can have an attribute that stores distribution of performance rating for the last ten years. We use a distribution descriptor to identify such tasks for tree visualizations. *Example*: "Identify distribution of the node value" [55].

4.5 Special Tasks

In our task dataset we identified tasks which had common conditions or instructions which give the task "special" context. An example of a special task is a memory task [50], where people are asked to find a node based on their memory recall. Special tasks are important because they reveal a niche problem for a particular dataset type. Furthermore users need a specialized strategy to perform these tasks. We identified four categories of special tasks in our survey as discussed below and illustrated with examples in **Fig. 6**.

Memory: Memory tasks for tree visualizations are designed to measure how well users can build a mental-map of the tree and find nodes they have previously viewed in another task or context. In the task "Return to Previously Visited Node" [54] users are asked to locate a node that they found in a previous node-finding task. The action classification of memory task is "Lookup" because the user is aware of the node position from the previous task.

Path: Path tasks are where the target of the task is a node, and the users have to follow a path to find the target. For instance, in the task "Find the directory yidemo (/hcil/lifelines/yidemo)" [49] the final target of the task is the "yidemo" directory and the task explicitly provides the entire path to the target. In other non-path-following node finding tasks, "Node search (e.g., find node entitled 'cannibalism')" [85], users are free to choose any strategy to identify the node of interest.

Non-hierarchical Relations: Non-hierarchical relation tasks require users to find non-hierarchical connections between tree elements. For example, in sports tournaments players advance through initial games and play-off rounds to quarter and semi-finals with the winner ending at the root of the hierarchy. Here, additional relations might include tracing players of a particular nationality, or discovering which players are playing at which facilities [66]. To show non-hierarchical relations in tree visualization, authors connect subtrees through a direct path [52], [56], [66], [95].

Common Ancestor: In a common ancestor task, users have to find an ancestor node in a tree that two or more descendants have in common. Common ancestor tasks are useful to identify an intermediate set of parents for two nodes. For instance, the theory of evolution states that all life on earth has a common ancestor [96]. However, some species like the chimpanzee and the gorilla are more closely related, thus have far more common ancestors than chimpanzee and rodent.

4.6 Curated Task Dataset Creation

In this paper, we contribute a list of 212 analytical tree visualization tasks surveyed from published literature. Our novel MLTT extension for tree visualizations facilitated the curation of the final tree tasks dataset. To create our task dataset, we manually abstracted each task from the literature survey (Sec. 3.4) and categorized each task with a set of actions (Sec. 4.1) and targets (Sec. 4.2). This included identifying if the task had a low-level target descriptor (Sec. 4.4) or if it belonged to one of the special task categories (Sec. 4.5). The task dataset enables researchers and practitioners to compare visualization tasks through the means of their abstraction, identify the state of the art in tree visualization evaluation, and explore patterns and trends in the way tasks are phrased in existing literature. We discuss use-cases for this task dataset in Sec. 5. The full curated task dataset is available in the Supplemental Material, at https://osf.io/u5ehs/, and publicly accessible on our project website https://intervis-projects.ccs.neu.edu/Tree-Visualization-Survey/.

Abstraction Validation: To create the task dataset, we followed a three-phase tagging task abstraction procedure. In the first phase, the lead author tagged each task in the database as a set of action items from MLTT and target items from the new MLTT tree extension. Following this, two co-authors of the paper examined the abstraction from the previous stage and flagged all the tasks with potential discrepancies. In the final stage, the authors discussed all the contentious tasks and agreed upon a final categorization. The validation process sparked debate on sequenced tasks. Next, we elaborate our abstraction decisions for sequenced tasks to ensure future applications of the MLTT tree extension avoid categorization confusion.

Action Classification For Sequenced Tasks: For some particular tree visualization tasks, users need to perform a sequence of actions to complete a single task. In the task "Which directory includes a deeper hierarchy: Flutes or Guitars?" [44], the final goal is the height comparison of two subtrees. However, to compare the subtrees, a user needs first to find them in the hierarchy. For the abstraction of such tasks, we initially broke down the tasks into granular unit tasks. However, the added granularity of the tasks skewed our dataset towards the identify action. To resolve this problem, for each multi-step task we identify the primary goal. In the task we discussed earlier, the primary goal of the task was the comparison of flutes and guitars subtrees. Factoring in the final goal of the task and intention of the researcher, we tag the task with the action that is closest to the primary purpose of the task.

Task Abstraction Tally: After the task abstraction step, we analyzed the trends in tree visualization tasks by counting the total number of tasks corresponding to the mid- and low-level action and three levels of targets. For targets, we use nested reporting because the target extension is a nested model. In the target results (Table 5), we see that tree, subtree, and path have a "-" (blank) target attribute. The blank field means the exact target attribute was not defined in the task. In the task: "Checking the existence of an identical sub-hierarchy elsewhere in the plot." [72], a user had to look for similar subtrees. However, the task did not define how to compare the similarity. The similarity can be on the height, balance, or any other structural or attribute characteristic. We also noticed that tree visualizations underutilized the links for representing data attributes. For instance, in the "attribute" targets, we did not find any task related to the link in the tree. These observations are discussed further in Sec. 6. The tally for mid- and low-level actions are included in the Supplemental Material.

Generic Tasks Generated from Abstraction Results: The extended MLTT typology enabled abstraction of domain-specific tree visualization tasks to domain-independent abstract concepts (see examples in Sec. 4.3). However, the abstraction does not summarize the different tree visualization task options in a way that is easy to read and independent of domain jargon. To provide our readers with such a list, we use abstract concepts and generate a list of generic human-readable formated tasks. Both tasks: "How many documents are available on the topic of The Governments of Thu¨ringen?" [86] and "How many detected pesticides in leafy vegetables in Tianjin city?" [69] have the exact same abstraction Action: {Lookup, Identify} and Target:{Topology, Node, Degree}. A generic version of these tasks can be "Identify the degree of a node in the tree." A list of generic tasks can help researchers design evaluations. The list can serve as a checklist for the researchers to identify the different types of tasks they can include in the evaluation. Researchers can use the generic task phrasing to develop domain specific tasks. The researchers have to replace abstract keywords like node or level with appropriate domain-specific terminology to translate generic tasks to domain-specific. We include an overview of common tree visualization tasks in a generic format corresponding to the tree visualization targets in Table. 4. The detailed list of generic tasks, where we provide example generic tasks for each combination of action and target, is included in the Supplemental Material.

5 USE-CASE SCENARIOS

Our novel MLTT extension for tree visualization tasks and curated tree task dataset have the ability to help visualization researchers and practitioners to describe a domain-centric tree visualization task as a well specified abstract tree visualization task, analyze the state of the art in tree visualization task evaluation, and discover appropriate tree visualization encodings. In this section, we present and discuss three use-case scenarios to highlight the utility of the MLTT extension and task dataset.

Abstract domain-specific tasks and communicate to a wider audience: Nanxi is a visualization researcher working on a research project to visualize the hierarchical blood flow pathways in the human brain. Her domain collaborators request her to create a visualization that can compare the blood flow between the right and left brain arteries. To strip down the domain-specific terminology from the task descriptions and treat it as a visualization problem, Nanxi uses the tree-specific MLTT extension for targets. Using the extended typology, Nanxi finds out that domain users want to identify if the left and right tree branches are balanced. Given the task abstraction, Nanxi can show the distribution of left and right branches using different tree visualization encodings to the domain users. Since there are many visual encoding options to present the data, Nanxi designs a user study to chose the most appropriate technique. The evaluation results show that traditional node-link visualizations were both accurate and well-perceived. Nanxi wants to communicate these results to people outside the domain of neurology. The task abstraction helps her with this communication as she can discuss in her final report the significance of her findings in the context of an abstract tree balance task. Our work enabled Nanxi to describe domain specific tasks abstractly and accurately for analysis and communication.

Design a tree visualization evaluation study: Jerome is a first-year Ph.D. graduate student interested in conducting an empirical evaluation study to compare visual encodings for developing design guidelines for tree visualizations. To identify areas of tree visualization research with few evaluation studies, i.e., under-explored research topics, he first identifies all the evaluation papers in our survey dataset, see Table. 3. Next he analyzes the methods, encodings, and tasks evaluated previously in the literature. Based on

Targets Tally

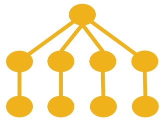
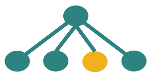
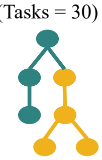
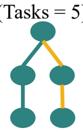
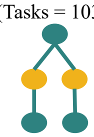
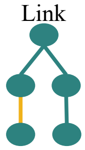
Target	Specific Target	Target Attribute	# of Tasks
Topology (Tasks = 109)	Tree (Tasks = 18) 	Balance	6
		Height	5
		Size	4
		-	2
		Order	1
		Fanout/Width	0
	Node (Tasks = 56) 	Degree	22
		Ancestors	14
		Ancestor/Descendant	8
		Depth	5
		Descendants	4
		Siblings	3
	Subtree (Tasks = 30) 	Size	20
		-	5
		Height	4
		Depth	1
	Path (Tasks = 5) 	-	4
		Length	1
Attribute (Tasks = 103)	Node (Tasks = 103) 	Categorical Value	56
		Quantitative Value	47
	Link 	Categorical Value	0
		Quantitative Value	0

TABLE 5
A tally of targets found in the survey dataset. For some tasks the target attribute was unresolvable therefore they are marked as "-". Further we found 8 tasks in "Node" that were related to finding ancestors or descendants but the task description was ambiguous therefore we categorize them as "Ancestor/Descendant". Corresponding to a few targets: Tree-Fanout(0) and Link(0), we did not find a task in our survey. This implies an opportunity and a need to study their effectiveness in the context of tree visualizations.

this analysis, Jerome learns that eye-tracking studies are an excellent method to identify visual search strategies and is an evaluation encoding not extensively used for tree visualizations in the literature. As a result, Jerome plans to conduct an eye-tracking study that evaluates tree visualization encodings and tasks not included in previous eye-tracking experiments [45], [48], [53], [76]. For evaluation tasks, Jerome can navigate through the tasks that have been used in previous eye-tracking studies, as the task information is available in the Supplemental Material. Based on his analysis, he can select visualization tasks that previous eye-tracking studies did not evaluate.

Explore visual encodings suitable for a task: Lisa is a visualization practitioner working at an artificial intelligence start-up. Lisa's team wants to visualize results of a hierarchical clustering algorithm. Through a series of interviews, Lisa identifies the following visualization requirement: identify the merge points of the different clusters. Lisa uses the tree-specific MLTT extension for targets to identify that that task is related to finding common ancestor nodes. Lisa uses the tree task dataset to identify visual encodings that have been used in the literature for the common ancestor tasks. She also is able to compile a comprehensive list of evaluation, technique, and application papers that have previously evaluated the tree visualization encodings for the common ancestor tasks. Lisa then plans to explore these research articles in detail to identify if she can use any of the existing visualization encodings for her project. In this use-case, our associated task dataset can act as an exploratory resource for practitioners to explore tree visualization encodings on the basis of task similarity.

It is important to note that we do not explicitly recommend in this paper nor in the task dataset any particular tree visualization given particular user goals. The creation of a full visual encoding recommendation system from evaluation results in the literature is a promising path for future work. In the mean time, we hope our task dataset will provide the necessary data to enable visualization researchers and practitioners to evaluate and design their tree visualizations.

6 Discussion & Future Work

Our tree visualization task survey, MLTT tree-specific extension, and task dataset creation were motivated by the lack of a comprehensive collection of tree visualization tasks and limited support by existing frameworks to support tree visualization task abstraction. To address these problems we contribute the first, and to the best of our knowledge only, task dataset as well as an extension to the MLTT framework to specifically abstract tree visualization tasks. In addition to these advancements, we also contribute in this section a discussion of open research questions identified in our literature survey and task dataset. These open questions relate to both tree visualization tasks and more generally to analytical tasks in other visualization encodings. Through this discussion, we emphasize how our work serves as a precursor to many potential future task-centered research agendas in the visualization community.

6.1 Opportunities for Tree Visualization Evaluations

Lack of crowdsourced studies: In our literature survey, we did not find a crowdsourced empirical evaluation that measures accuracy or time for human-subjects to complete analytical tasks with different tree visualization encodings. Even though we found one crowdsourced study that included tree visualizations [97], the study did not report analytical task driven analysis of accuracy and time measures. Unlike a controlled lab study, crowdsourced studies enable the evaluation of a visualization encoding using a large and diverse group of participants [98]. A large scale crowdsourced study can provide higher statistical power to analyze the results of the evaluation studies, and thus build more reliable knowledge about accuracy and completion times of a visualization encoding. There are also now available methods to replicate gaze-fixation data, as would be collected with traditional eye-tracking lab studies, in online crowd-sourced studies [99], [100]. We believe there is great potential for the visualization community to explore crowdsourced study options to increase the efficiency and efficacy of tree visualization studies.

Experimental design varies across studies: After the survey of tasks, we explored the empirical studies to identify general trends and insights which can help us develop guidelines to predict the most suitable tree visualization encoding for a particular type of task. However, in the meta-analysis of the empirical studies, we found that the experimental stimuli, methods and data analysis varied across the studies. The tree visualization stimuli used across studies differ significantly in design and features. For instance, around 30% of papers used static tree visualizations, and 70% used interactive visualizations. There is also variation in the statistics reported. While most of the empirical papers reported accuracy and time, two papers do not report accuracy [27], [49]. The dataset used in the evaluation can also affect the results. Our survey found that there is a certain degree of variation in the type of dataset being used in empirical evaluations. The two main types of datasets used are file-system [28], [42], [44], [47], [49], [60], [88] and biological evolution or genealogy datasets [46], [59], [64], [77], [79]. In contrast to studies that used datasets from a particular domain, some studies used abstract tasks without relating to any particular domain [43], [48]. The variations in experimental design also make comparison of empirical results challenging across studies. Saket et al. [36] discussed similar problems; they surveyed basic visualization design like bar charts and scatterplots and found that actual evaluation results are not comparable. Consequently, the authors had to conduct a novel empirical evaluation to study the effect of visualization design on task accuracy and time. Tree visualization research can also benefit from a holistic evaluation study that evaluates the shortcomings of existing empirical studies and designs an experiment that systematically studies the effect of visualization encoding on analytical tasks. In summary, the variations across experimental studies inhibits the process of knowledge gathering, and as a community we need to develop heuristics to enable comparison and guidelines to ensure researchers can design evaluations and communicate results in a consistent manner.

Lack of "enjoyment" measures in evaluations: At the high-level of the "action" classification, the MLTT framework (Sec. 4.1) defines an "enjoy" goal that refers to casual encounters with a visualization. It has been demonstrated that enjoyable visualizations lead to higher engagement, and users spend prolonged time with the visualizations [101]. During our survey, we found only a single paper [97] with the core focus on measuring the "enjoy" goals of a visualization. A fundamental question that emerges from this analysis is: *How can visualization researchers design evaluations to genuinely measure non-objective goals relevant to enjoyment such as "engagement" and "playfulness"?* Research in the visualization community on the evaluation of user-engagement is still in the early phases [101], and we believe specialized encodings including tree visualizations can significantly benefit from further development in this area.

Adjacency diagrams are a promising tree visualization method: The two primary types of tree visualization encodings in visualization literature are node-link and enclosure diagrams (Fig. 3). However, we found many papers that demonstrate the advantage of adjacency diagrams over the node-link and enclosure diagram. Stasko et al. [4] compared a sunburst chart (adjacency diagram) to a treemap (enclosure diagram) and found that for both structural and topology tasks the adjacency diagrams outperformed the enclosure diagram. In empirical studies [43], [48], authors show that adjacency diagrams are better at both structural and topology tasks than the node-link diagrams. Furthermore, an aesthetic evaluation of tree visualization encodings [97] showed that the sunburst chart was considered to be the most aesthetically pleasing tree visualization encoding. The empirical trends of the adjacency diagram coupled with the overall more space-efficient layout than the node-link encoding and explicit hierarchy representation make it a practical choice for tree visualization.

6.2 Tree Visualization Survey Extensions

Tree Surveys Beyond Tasks: Our survey primarily focuses on the analytical tasks of tree visualization, or the "Why do users need visualization?" in the MLTT framework. The extensive coverage of "Why" was intentional, given the lack of available resources for tree visualization tasks. With the detailed "Why" part of the framework now expanded, as a part of future work tree visualization encodings can be analyzed individually and described as a set across the full MLTT typology: "*What* is the underlying data model?", "*Why* was the visualization designed?", and "*How* was the data encoded?". We believe classification of existing tree visualizations based on their data, task, and encoding can empower visualization researchers and will be an important step towards formalizing visualization knowledge for tree visualizations.

Extension of Task Dataset: The current task dataset contains 212 tree visualization tasks and their abstraction. In future work, we aim to develop a system for community members to add tasks to the dataset or edit existing task abstraction. The addition of tasks can be community-driven but will require authorization and validation from an expert researcher. In this case, the expert researchers will be the authors of this work or community members who have expertise in tree visualization and task abstraction research. We also want to enable a feature for community members to contest an abstraction. The ability to request abstraction edits will help the community better understand how to abstract tree visualization tasks. Adding new tasks to the task dataset will ensure that the task dataset remains relevant as the literature on tree visualization tasks and evaluation expands.

6.3 Task Datasets Beyond Tree Visualizations

We advocate for the creation of curated task datasets for other data types such as network, geospatial, and specialized fields/geometries. This paper presents a methodology to create a task dataset for tree visualizations, but the process can be applied to other data types. Based on our experience, we argue for the continued use of MLTT for dataset-specific target extensions for the curation of these task datasets. The MLTT framework classifies user actions for tree visualization tasks, and its set of meticulously selected "action" keywords delineate the objectives at each stage of the task. As with our tree extension, a nested target extension can augment the action classification of MLTT. The target extension is particularly useful for data types where researchers are not able to specify the target to a level that it can be uniquely identified. By working together as a community we can collectively extend the framework to the benefit of all.

6.4 Challenges and Opportunities for Task Research

Inconsistencies in Task Phrasing: In our literature survey, we found that the current language of tree visualization tasks lack consistent phrasing. We identified two common problems with the phrasing of tasks in the current literature: **1. Lack of standardized terminology for tasks:** The terminology inconsistency across publications leads to the use of varying terms for the same tree visualization attribute. For example, Kosba et al. [44] phrased a tree height task as a depth lookup task: "What is the maximum **depth** of the eBay hierarchy?". In another study, Ham et al. [50] phrased a tree height task in terms of levels: "Users had to indicate the total number of **levels** in the tree." Even though both tasks are tied to the same target (height), they not only look very different because of the terminology but could be incorrectly interpreted or applied by other researchers. **2. Inconsistent phrasing of tasks:** Another problem we observed in the literature is the lack of consistent phrasing of tree visualization tasks. To give an example of this inconsistency, consider the following two tasks: "Estimating which of two sub-hierarchies was the larger one" [72] and "The subjects were instructed to compare the two directories and select the one that contained most descendants" [78]. In both tasks, the main goal is the comparison of two subtrees. The first task is phrased casually and is less indicative of what the researcher wants the user to achieve. It casually mentions find the "larger" sub-hierarchy. However, it does not explain what exactly "larger" means in the context of tree visualization. On the other hand, the second task is clear in what it expects from the user: they want to identify the subtree with the "most descendants."

These inconsistencies in tree visualization task phrasing can inhibit the comparison of empirical evidence from different studies by impeding the process of knowledge generation for building design guidelines for visualization practitioners. To reduce task phrasing inconsistencies, we need to develop a standard method for phrasing and reporting tree visualization tasks. We argue that consistency in task description can be improved by using keywords from the task abstraction theory. For instance, tasks like "Find the node with the highest value" or "Look for a node that is largest in size" could be more consistently phrased as "Identify the node with maximum value". In the Supplemental Material, we propose an abstract definition for tree tasks constructed with standard vocabulary from MLTT and our tree-specific extension to MLTT and discuss how the task abstraction can be improved and validated for more widespread usage in the visualization community.

Building task abstraction guidelines: Task abstraction enables the comparison of tasks from a wide range of papers through the use of a standard language structure and vocabulary. However, as we discuss in this paper, we had to create a new framework and vocabulary for tree task abstraction. In order to use MLTT for trees, we had to elaborate on the general terms used in the typology and to add tree visualization specific context. For instance, to classify a task as "lookup" or "browse" we need information on the "location" of the target. "Location" in the typology has an open-ended meaning, and it can mean an actual physical position on a chart or a conceptual location like adjacent to an item. Therefore, to abstract tree visualization tasks, we had to add specificity to the meaning of "location" (discussed in Sec. 4.1). Ultimately, if task datasets are created for other visual encodings like network or spatial visualizations, then a byproduct of that work will be a set of guidelines to elaborate on the meaning of MLTT definitions and terminology in the context of the specialized visual encoding. These additions of specificity to the framework and extensions for other visual encodings will create a more robust framework for tasks abstraction and enable more effective visualization research and creation.

Towards intelligent abstraction: The manual task abstraction process discussed in Sec. 4.6 is prone to errors originating with the visualization researchers including biases coming from previous work experiences and a possible lack of domain knowledge which could lead to task/goal mismatches. In light of these possible errors introduced into the task abstraction process, there is an opportunity for an artificial intelligence system to help visualization practitioners by augmenting their ability to validate and reason about the task abstraction [102]. In order to create an intelligent system to perform task abstraction, we need to train machine learning models with real tasks and their labeled outputs. Task datasets, such as the one contributed in this paper, could provide the necessary training data to build such models to perform task abstraction. The visualization community has begun to formalize its efforts to understand the potential for AI to support visualization research and design [103], and we see the application of intelligent systems to task abstraction as a rich area for research.

7 Conclusions

Tasks are an integral part of the visualization design process. Without a proper understanding of analytical tasks, visualization creators are bound to make design errors thus limiting a user's ability to read and interpret a visualization. In this paper we present a collection of 212 tree visualization tasks and a novel extension of the Multi-Level Task Typology (MLTT) to accurately abstract and analyze the task collection. In order to collect tree visualization tasks, we surveyed 54 tree visualization papers across a range of paper types with domains varying from biology to geography to social science. The survey enabled us to build a novel extension to the MLTT for trees, which includes an exhaustive list of tree visualization targets, and enabled us to conduct the systematic abstraction of surveyed tasks. The meta-analysis of this abstraction process contributes a set of guidelines for practitioners and researchers to perform abstraction of tree visualization tasks in the future. The survey also enabled us to identify open research questions related to the phrasing of analytical tasks in the existing literature and evaluation shortcomings of the current tree visualization literature. Through these contributions we hope that visualization practitioners will gain a better understanding of tree visualization tasks, and that researchers outside of tree visualization will perform similar research to enhance the understanding of analytical tasks for other dataset types.

Acknowledgments

The authors wish to thank Hans-Jörg Schulz for initial feedback on the project idea. We are also thankful to Yixuan "Janice" Zhang, Cody Dunne, Micha Schwab, David Saffo, Sara Di Bartolomeo, Laura South and other members of visualization lab at Northeastern University for their feedback on the paper. We are also grateful to Amriteya Pandey for contributing to our website development. Finally, we thank Jane Kokernak for her help with editing of the paper. This work was supported by NSF CISE CRII award no. 1657466.

References

[1] T. Munzner, *Visualization analysis and design*. CRC press, 2014.

[2] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2376–2385, Dec 2013.

[3] M. Sedlmair, M. Meyer, and T. Munzner, "Design study methodology: Reflections from the trenches and the stacks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2431–2440, Dec 2012.

[4] J. Stasko, R. Catrambone, M. Guzdial, and K. McDonald, "An evaluation of space-filling information visualizations for depicting hierarchical structures," *International journal of human-computer studies*, vol. 53, no. 5, pp. 663–694, 2000.

[5] L. Yu, P. Svetachov, P. Isenberg, M. H. Everts, and T. Isenberg, "Fi3d: Direct-touch interaction for the exploration of 3d scientific visualization spaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1613–1622, Nov 2010.

[6] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale, "Empirical studies in information visualization: Seven scenarios," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1520–1536, Sep. 2012.

[7] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry, "Task taxonomy for graph visualization," in *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, ser. BELIV '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 1–5.

[8] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers, "Visualizing Sets and Set-typed Data: State-of-the-Art and Future Challenges," in *EuroVis - STARs*, R. Borgo, R. Maciejewski, and I. Viola, Eds. The Eurographics Association, 2014.

[9] M. Meyer, T. Munzner, and H. Pfister, "Mizbee: A multiscale synteny browser," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 897–904, Nov 2009.

[10] S. Nusrat and S. Kobourov, "Task Taxonomy for Cartograms," in *Eurographics Conference on Visualization (EuroVis) - Short Papers*, E. Bertini, J. Kennedy, and E. Puppo, Eds. The Eurographics Association, 2015.

[11] R. Amar, J. Eagan, and J. Stasko, "Low-level components of analytic activity in information visualization," in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, Oct 2005, pp. 111–117.

[12] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE Symposium on Visual Languages*, 1996, pp. 336–343.

[13] J. Fuchs, P. Isenberg, A. Bezerianos, and D. Keim, "A systematic review of experimental studies on data glyphs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1863–1879, July 2017.

[14] S. Nusrat, T. Harbig, and N. Gehlenborg, "Tasks, techniques, and tools for genomic data visualization," *Computer Graphics Forum*, vol. 38, no. 3, pp. 781–805, 2019.

[15] S. Liu, X. Wang, C. Collins, W. Dou, F. Ouyang, M. El-Assady, L. Jiang, and D. A. Keim, "Bridging text visualization and mining: A task-driven survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 7, pp. 2482–2504, July 2019.

[16] N. Kerracher, J. Kennedy, and K. Chalmers, "A task taxonomy for temporal graph visualisation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 10, pp. 1160–1172, Oct 2015.

[17] G. Andrienko, N. Andrienko, P. Bak, D. Keim, S. Kisilevich, and S. Wrobel, "A conceptual framework and taxonomy of techniques for analyzing movement," *Journal of Visual Languages & Computing*, vol. 22, no. 3, pp. 213–232, 2011.

[18] H. Schulz, T. Nocke, M. Heitzler, and H. Schumann, "A design space of visualization tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2366–2375, Dec 2013.

[19] M. Lima, *The book of trees: visualizing branches of knowledge*. Princeton Architectural Press, 2014.

[20] I. Herman, G. Melancon, and M. S. Marshall, "Graph visualization and navigation in information visualization: A survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 24–43, Jan 2000.

[21] B. Shneiderman, "Tree visualization with tree-maps: 2-d space-filling approach," *ACM Trans. Graph.*, vol. 11, no. 1, p. 92–99, Jan. 1992.

[22] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. van Wijk, J.-D. Fekete, and D. Fellner, "Visual analysis of large graphs: State-of-the-art and future research challenges," *Computer Graphics Forum*, vol. 30, no. 6, pp. 1719–1749, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2011.01898.x

[23] I. Letunic and P. Bork, "Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation," *Bioinformatics*, vol. 23, no. 1, pp. 127–128, 10 2006.

[24] J.-D. Fekete and C. Plaisant, "General Tasks Applicable to Most Trees," https://www.cs.umd.edu/hcil/iv03contest/generaltasks.html, 2002.

[25] H. Schulz, "Treevis.net: A tree visualization reference," *IEEE Computer Graphics and Applications*, vol. 31, no. 6, pp. 11–15, Nov 2011.

[26] T. Munzner, "Interactive visualization of large graphs and networks," Ph.D. dissertation, Stanford University, 2000.

[27] A. Cockburn and B. McKenzie, "An evaluation of cone trees," in *People and Computers XIV — Usability or Else!*, S. McDonald, Y. Waern, and G. Cockton, Eds. London: Springer London, 2000, pp. 425–436.

[28] T. Bladh, D. A. Carr, and J. Scholl, "Extending tree-maps to three dimensions: A comparative study," in *Computer Human Interaction*, M. Masoodian, S. Jones, and B. Rogers, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 50–59.

[29] J. B. Kruskal and J. M. Landwehr, "Icicle plots: Better displays for hierarchical clustering," *The American Statistician*, vol. 37, no. 2, pp. 162–168, 1983.

[30] Shengdong Zhao, M. J. McGuffin, and M. H. Chignell, "Elastic hierarchies: combining treemaps and node-link diagrams," in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, Oct 2005, pp. 57–64.

[31] E. Kleiberg, H. van de Wetering, and J. J. Van Wijk, "Botanical visualization of huge hierarchies," in *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, ser. INFOVIS '01. USA: IEEE Computer Society, 2001, p. 87.

[32] M. Graham and J. Kennedy, "A survey of multiple tree visualisation," *Information Visualization*, vol. 9, no. 4, p. 235–252, Dec. 2010. [Online]. Available: https://doi.org/10.1057/ivs.2009.29

[33] G. Li, Y. Zhang, Y. Dong, J. Liang, J. Zhang, J. Wang, M. J. Mcguffin, and X. Yuan, "Barcodetree: Scalable comparison of multiple hierarchies," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 1022–1032, Jan 2020.

[34] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou, "Treejuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility," in *ACM SIGGRAPH 2003 Papers*, ser. SIGGRAPH '03. New York, NY, USA: Association for Computing Machinery, 2003, p. 453–462.

[35] S. K. Card, B. Suh, B. A. Pendleton, J. Heer, and J. W. Bodnar, "Time tree: Exploring time changing hierarchies," in *2006 IEEE Symposium On Visual Analytics Science And Technology*, Oct 2006, pp. 3–10.

[36] B. Saket, A. Endert, and C. Demiralp, "Task-based effectiveness of basic visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 7, pp. 2505–2512, July 2019.

[37] A. Ottley, R. J. Crouser, C. Ziemkiewicz, and R. Chang, "Manipulating and controlling for personality effects on visualization tasks," *Information Visualization*, vol. 14, no. 3, pp. 223–233, 2015.

[38] B. Johnson and B. Shneiderman, "Tree-maps: a space-filling approach to the visualization of hierarchical information structures," in *Proceedings Visualization '91*, Oct 1991, pp. 284–291.

[39] M. Harrigan and P. Healy, "Efficiently drawing a significant spanning tree of a directed graph," in *2007 6th International Asia-Pacific Symposium on Visualization*, Feb 2007, pp. 53–59.

[40] K. Kucher and A. Kerren, "Text visualization techniques: Taxonomy, visual survey, and community insights," in *2015 IEEE Pacific Visualization Symposium (PacificVis)*, April 2015, pp. 117–121.

[41] L. O. Hall, N. Chawla, and K. W. Bowyer, "Decision tree learning on very large data sets," in *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, vol. 3, Oct 1998, pp. 2579–2584 vol.3.

[42] B. S. Johnson, "Treemaps: visualizing hierarchical and categorical data," Ph.D. dissertation, University of Maryland, 1993.

[43] T. Barlow and P. Neville, "A comparison of 2-d visualizations of hierarchies," in *IEEE Symposium on Information Visualization, 2001. INFOVIS 2001.*, 2001, pp. 131–138.

[44] A. Kobsa, "User experiments with tree visualization systems," in *IEEE Symposium on Information Visualization*, Oct 2004, pp. 9–16.

[45] M. Burch, N. Konevtsova, J. Heinrich, M. Hoeferlin, and D. Weiskopf, "Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2440–2448, 2011.

[46] C. Plaisant, J. Grosjean, and B. B. Bederson, "Spacetree: supporting exploration in large node link tree, design evolution and empirical evaluation," in *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, 2002, pp. 57–64.

[47] Y. Wang, S. T. Teoh, and K.-L. Ma, "Evaluating the effectiveness of tree visualization systems for knowledge discovery," in *Proceedings of the Eighth Joint Eurographics / IEEE VGTC Conference on Visualization*, ser. EUROVIS'06. Goslar, DEU: Eurographics Association, 2006, p. 67–74.

[48] N. H. Müller, B. Liebold, D. Pietschmann, P. Ohler, and P. Rosenthal, "Hierarchy visualization designs and their impact on perception and problem solving strategies," in *ACHI 2017*, 2017.

[49] K. Andrews and J. Kasanicka, "A comparative study of four hierarchy browsers using the hierarchical visualisation testing environment (hvte)," in *2007 11th International Conference Information Visualization (IV '07)*, 2007, pp. 81–86.

[50] F. van Ham and J. J. van Wijk, "Beamtrees: compact visualization of large hierarchies," in *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, 2002, pp. 93–100.

[51] C. Ziemkiewicz and R. Kosara, "The shaping of information by visual metaphors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1269–1276, 2008.

[52] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 741–748, 2006.

[53] L. R. Novick, A. T. Stull, and K. M. Catley, "Reading Phylogenetic Trees: The Effects of Tree Orientation and Text Processing on Comprehension," *BioScience*, vol. 62, no. 8, pp. 757–764, 08 2012.

[54] T. D. Wang and B. Parsia, "Cropcircles: Topology sensitive visualization of owl class hierarchies," in *The Semantic Web - ISWC 2006*, I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. M. Aroyo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 695–708.

[55] J. Görtler, C. Schulz, D. Weiskopf, and O. Deussen, "Bubble treemaps for uncertainty visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 719–728, Jan 2018.

[56] T. Dang and A. Forbes, "Cactustree: A tree drawing approach for hierarchical edge bundling," in *2017 IEEE Pacific Visualization Symposium (PacificVis)*, 2017, pp. 210–214.

[57] R. Blanch, R. Dautriche, and G. Bisson, "Dendrogramix: A hybrid tree-matrix visualization technique to support interactive exploration of dendrograms," in *2015 IEEE Pacific Visualization Symposium (PacificVis)*, April 2015, pp. 31–38.

[58] D. Auber, C. Huet, A. Lambert, B. Renoust, A. Sallaberry, and A. Saulnier, "Gospermap: Using a gosper curve for laying out hierarchical data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 11, pp. 1820–1832, 2013.

[59] F. Block, M. S. Horn, B. C. Phillips, J. Diamond, E. M. Evans, and C. Shen, "The deeptree exhibit: Visualizing the tree of life to facilitate informal learning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2789–2798, 2012.

[60] L. Linsen and S. Behrendt, "Linked treemap: a 3d treemap-nodelink layout for visualizing hierarchical structures," *Computational Statistics*, vol. 26, no. 4, p. 679, 2011.

[61] C. Tuttle, L. G. Nonato, and C. Silva, "Pedvis: A structured, space-efficient technique for pedigree visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1063–1072, 2010.

[62] H. Song, B. Kim, B. Lee, and J. Seo, "A comparative evaluation on tree visualization methods for hierarchical structures with large fan-outs," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 223–232.

[63] J. Guerra-Gómez, M. L. Pack, C. Plaisant, and B. Shneiderman, "Visualizing change over time using dynamic hierarchies: Treeversity2 and the stemview," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2566–2575, Dec 2013.

[64] R. Therón, "Hierarchical-temporal data visualization using a tree-ring metaphor," in *Smart Graphics*, A. Butz, B. Fisher, A. Krüger, and P. Olivier, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 70–81.

[65] D. Tan, G. Smith, B. Lee, and G. Robertson, "Adaptivitree: Adaptive tree visualization for tournament-style brackets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1113–1120, 2007.

[66] P. Neumann, D. S. Schlechtweg, and S. Carpendale, "ArcTrees: Visualizing Relations in Hierarchical Data," in *EUROVIS 2005: Eurographics / IEEE VGTC Symposium on Visualization*, K. Brodlie, D. Duke, and K. Joy, Eds. The Eurographics Association, 2005.

[67] U. Wiss, D. Carr, and H. Jonsson, "Evaluating three-dimensional information visualization designs: a case study of three designs," in *Proceedings. 1998 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics (Cat. No.98TB100246)*, 1998, pp. 137–144.

[68] A. G. M. Soares, D. H. dos Santos, C. L. R. Barbosa, A. S. Gonçalves, C. G. R. dos Santos, B. S. Meiguins, and E. T. C. Miranda, "Visualizing multidimensional data in treemaps with adaptive glyphs," in *2018 22nd International Conference Information Visualisation (IV)*, 2018, pp. 58–63.

[69] Y. Chen, X. Zhang, Y. Feng, J. Liang, and H. Chen, "Sunburst with ordered nodes based on hierarchical clustering: a visual analyzing method for associated hierarchical pesticide residue data," *Journal of Visualization*, vol. 18, no. 2, pp. 237–254, 2015.

[70] S. van den Elzen and J. J. van Wijk, "Baobabview: Interactive construction and analysis of decision trees," in *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2011, pp. 151–160.

[71] M. Wattenberg and F. B. Viégas, "The word tree, an interactive visual concordance," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1221–1228, 2008.

[72] M. Burch, M. Raschke, and D. Weiskopf, "Indented pixel tree plots," in *Proceedings of the 6th International Conference on Advances in Visual Computing - Volume Part I*, ser. ISVC'10. Berlin, Heidelberg: Springer-Verlag, 2010, p. 338–349.

[73] A. Sallaberry, Y.-c. Fu, H.-C. Ho, and K.-L. Ma, "Correction: Contact trees: Network visualization beyond nodes and edges," *PLOS ONE*, vol. 11, no. 2, pp. 1–1, 02 2016.

[74] L. Woodburn, Y. Yang, and K. Marriott, "Interactive visualisation of hierarchical quantitative data: An evaluation," in *2019 IEEE Visualization Conference (VIS)*. IEEE, 2019, pp. 96–100.

[75] J. Beheshti, A. Large, C.-A. Julien, and M. Tam, "A comparison of a conventional taxonomy with a 3d visualization for use by children," in *Proceedings of the 73rd ASIST Annual Meeting on Navigating Streams in an Information Ecosystem - Volume 47*, ser. ASIST '10. USA: American Society for Information Science, 2010.

[76] V. F. de Santana, F. M. de Moraes, and B. S. R. de Souza, "Investigating treemap visualization in inverted scale," in *Proceedings of the 14th Brazilian Symposium on Human Factors in Computing Systems*, ser. IHC '15. New York, NY, USA: Association for Computing Machinery, 2015.

[77] T. Bladh, D. A. Carr, and M. Kljun, "The effect of animated transitions on user navigation in 3d tree-maps," in *Ninth International Conference on Information Visualisation (IV'05)*, 2005, pp. 297–305.

[78] U. Wiss and D. A. Carr, "An empirical study of task support in 3d information visualizations," in *1999 IEEE International Conference on Information Visualization (Cat. No. PR00210)*, 1999, pp. 392–399.

[79] J. M. Santos, B. S. Santos, P. Dias, S. Silva, and C. Ferreira, "Extending the h-tree layout pedigree: An evaluation," in *2013 17th International Conference on Information Visualisation*, 2013, pp. 422–427.

[80] R. P. Biuk-Aghai, P. C.-I. Pang, and B. Pang, "Map-like visualisations vs. treemaps: An experimental comparison," in *Proceedings of the 10th International Symposium on Visual Information Communication and Interaction*, ser. VINCI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 113–120.

[81] J. Liang, Q. V. Nguyen, S. Simoff, and M. L. Huang, "Angular treemaps - a new technique for visualizing and emphasizing hierarchical structures," in *2012 16th International Conference on Information Visualisation*, 2012, pp. 74–80.

[82] H. v. de Wetering, N. Klaassen, and M. Burch, "Space-reclaiming icicle plots," in *2020 IEEE Pacific Visualization Symposium (PacificVis)*, June 2020, pp. 121–130.

[83] G. Li-Wei, C. Yi, Z. Xin-Yue, and S. Yue-Hong, "A hierarchical data visualization algorithm: Self-adapting sunburst algorithm," in *2013 International Conference on Virtual Reality and Visualization*, 2013, pp. 185–190.

[84] H. Shin, G. Park, and J. Han, "Tablorer – an interactive tree visualization system for tablet pcs," *Computer Graphics Forum*, vol. 30, no. 3, pp. 1131–1140, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2011.01962.x

[85] Z. Band and R. W. White, "Pygmybrowse: A small screen tree browser," in *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 514–519.

[86] K. Andrews, W. Kienreich, V. Sabol, J. Becker, G. Droschl, F. Kappe, M. Granitzer, P. Auer, and K. Tochtermann, "The infosky visual explorer: Exploiting hierarchical structure and document similarities," *Information Visualization*, vol. 1, no. 3-4, pp. 166–181, 2002.

[87] T. Dang, P. Murray, R. Etemadpour, and A. G. Forbes, "A user study of techniques for visualizing structure and connectivity in hierarchical datasets."

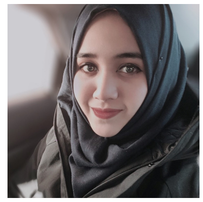
[88] S. Muramalla, R. Al Tarawneh, S. R. Humayoun, R. Moses, S. Panis, and A. Ebert, "Radial vs. rectangular: Evaluating visualization layout impact on user task performance of hierarchical data." *IADIS International Journal on Computer Science & Information Systems*, vol. 12, no. 2, 2017.

[89] M. Golemati, A. Katifori, E. G. Giannopoulou, I. Daradimos, and C. Vassilakis, "Evaluating the significance of the windows explorer visualization in personal information management browsing tasks," in *2007 11th International Conference Information Visualization (IV '07)*, July 2007, pp. 93–100.

[90] L. K. Long, L. C. Hui, G. Y. Fook, and W. M. N. Wan Zainon, "A study on the effectiveness of tree-maps as tree visualization techniques," *Procedia Computer Science*, vol. 124, pp. 108 – 115, 2017, 4th Information Systems International Conference 2017, ISICO

2017, 6-8 November 2017, Bali, Indonesia. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050917329046
[91] A. Heinicke, C. Liao, K. Walbaum, J. Bützler, and C. M. Schlick, "User centered evaluation of interactive data visualization forms for document management systems," *Procedia Manufacturing*, vol. 3, pp. 5427 – 5434, 2015, 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2351978915006708
[92] H. Heath and S. Cowley, "Developing a grounded theory approach: a comparison of glaser and strauss," *International Journal of Nursing Studies*, vol. 41, no. 2, pp. 141 – 150, 2004.
[93] D. Holten and J. J. van Wijk, "Visual comparison of hierarchically organized data," in *Proceedings of the 10th Joint Eurographics / IEEE - VGTC Conference on Visualization*, ser. EuroVis'08. Chichester, GBR: The Eurographs Association & John Wiley & Sons, Ltd., 2008, p. 759–766.
[94] E. M. Reingold and J. S. Tilford, "Tidier drawings of trees," *IEEE Transactions on software Engineering*, no. 2, pp. 223–228, 1981.
[95] F. Viégas, M. Wattenberg, J. Hebert, G. Borggaard, A. Cichowlas, J. Feinberg, J. Orwant, and C. Wren, "Google+ripples: A native visualization of information flow," in *Proceedings of the 22nd International Conference on World Wide Web*, ser. WWW '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 1389–1398.
[96] J. Rosindell and L. J. Harmon, "Onezoom: A fractal explorer for the tree of life," *PLOS Biology*, vol. 10, no. 10, pp. 1–5, 10 2012.
[97] N. Cawthon and A. V. Moere, "The effect of aesthetic on the usability of data visualization," in *2007 11th International Conference Information Visualization (IV '07)*, 2007, pp. 637–648.
[98] R. Borgo, B. Lee, B. Bach, S. Fabrikant, R. Jianu, A. Kerren, S. Kobourov, F. McGee, L. Micallef, T. von Landesberger, K. Ballweg, S. Diehl, P. Simonetto, and M. Zhou, "Crowdsourcing for information visualization: Promises and pitfalls," in *Evaluation in the Crowd. Crowdsourcing and Human-Centered Experiments*, D. Archambault, H. Purchase, and T. Hoßfeld, Eds. Cham: Springer International Publishing, 2017, pp. 96–138.
[99] N. W. Kim, Z. Bylinskii, M. A. Borkin, K. Z. Gajos, A. Oliva, F. Durand, and H. Pfister, "Bubbleview: An interface for crowdsourcing image importance maps and tracking visual attention," *ACM Trans. Comput.-Hum. Interact.*, vol. 24, no. 5, Nov. 2017.
[100] S. R. Gomez, R. Jianu, R. Cabeen, H. Guo, and D. H. Laidlaw, "Fauxvea: Crowdsourcing gaze location estimates for visualization analysis tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 2, pp. 1042–1055, 2017.
[101] N. Mahyar, S.-H. Kim, and B. C. Kwon, "Towards a taxonomy for evaluating user engagement in information visualization," 2015.
[102] A. Pandey, Y. Zhang, J. A. Guerra-Gomez, A. G. Parker, and M. A. Borkin, "Digital collaborator: Augmenting task abstraction in visualization design with artificial intelligence," 2020.
[103] M. Vartak, S. Huang, T. Siddiqui, S. Madden, and A. Parameswaran, "Towards visualization recommendation systems," *SIGMOD Rec.*, vol. 45, no. 4, p. 34–39, May 2017.

Aditeya Pandey is a Ph.D. Candidate in the Khoury College of Computer Sciences at Northeastern University, Boston, MA, US. His research interests are data visualization for network and hierarchical datasets, developing recommender systems for data visualization, and quantitative and qualitative evaluation of visualization designs.



Uzma Haque Syeda received her bachelor's degree in Electrical and Electronic Engineering from The University of Dhaka, Bangladesh in 2017. Currently, she is pursuing a Ph.D. in Computer Science at Northeastern University, Boston, Massachusetts.

Chaitya Shah Chaitya Shah received his B. Tech in Computer Engineering from Dharmsinh Desai University, India and MS in Data Science from Northeastern University, Boston, MA. Currently, he is a Data Analyst with the Enterprise Risk Management group at Fidelity Investments in Boston, MA. His interests include NLP, anomaly detection, data visualization, and analyzing financial data.

John A. Guerra-Gomez received his Ph.D. in Computer Science from University of Maryland, College Park, Maryland, US. He is currently an Assisting Teaching Professor at Khoury College of Computer Sciences at Northeastern University, Silicon Valley Campus, CA, US. His interests includes Visual Analytics, Accessibility, Big data, Human Computer Interaction and Web Development.

Michelle A. Borkin received a Ph.D. in 2014 and an M.S. in 2011, both in Applied Physics, from Harvard University in Cambridge, MA, USA. She is an Assistant Professor in the Khoury College of Computer Sciences at Northeastern University, Boston, MA, USA. Her research interests include data visualization, human-computer interaction, and application work across domains including astronomy, physics, and medical imaging.

Evaluating the Effect of Timeline Shape on Visualization Task Performance

Sara Di Bartolomeo, Aditeya Pandey, Aristotelis Leventidis, David Saffo, Uzma Haque Syeda, Elin Carstensdottir, Magy Seif El-Nasr, Michelle A. Borkin, Cody Dunne
Northeastern University
(dibartolomeo.s | pandey.ad | saffo.d | syeda.u)@husky.neu.edu, elin@ccs.neu.edu,
(magy | m.borkin | c.dunne)@northeastern.edu

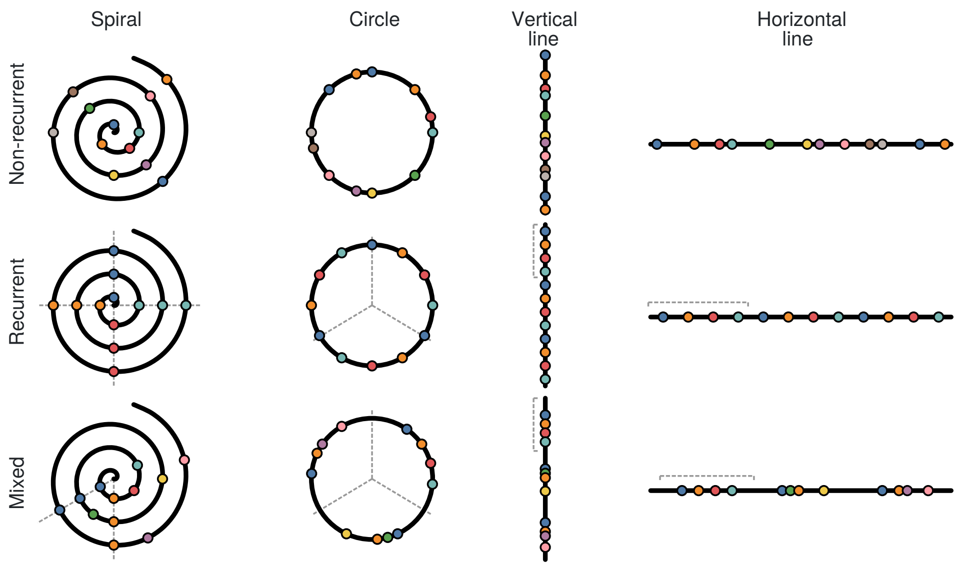


Figure 1. We evaluate the effect on task performance of 4 timeline shapes (left to right) across 3 types of temporal event sequence data (top to bottom). These images are simplified versions of the stimuli that we used in our experiment. Each dot on a timeline represents an event and has a specific categorical color to highlight where the dataset has recurrent events. Dashed lines highlight the recurrent intervals or a set of recurrent events.

ABSTRACT

Timelines are commonly represented on a horizontal line, which is not necessarily the most effective way to visualize temporal event sequences. However, few experiments have evaluated how timeline shape influences task performance. We present the design and results of a controlled experiment run on Amazon Mechanical Turk ($n = 192$) in which we evaluate how timeline shape affects task completion time, correctness, and user preference. We tested 12 combinations of 4 shapes — horizontal line, vertical line, circle, and spiral — and 3 data types — recurrent, non-recurrent, and mixed event sequences. We found good evidence that timeline shape meaningfully affects user task completion time but not correctness and that users have a strong shape preference. Building on our results, we present design guidelines for creating effective timeline visualizations based on user task and data types. A free copy of this paper, the evaluation stimuli and data, and code are available at https://osf.io/qr5yu/

Author Keywords

Timelines; Temporal Event Sequences; Information Visualization; Controlled Experiments

CCS Concepts

•**Human-centered computing** → **Human computer interaction (HCI);** *Visualization design and evaluation;* Information visualization;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CHI '20, April 25–30, 2020, Honolulu, HI, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6708-0/20/04 ...$15.00.
http://dx.doi.org/10.1145/3313831.3376237

INTRODUCTION

A timeline is a visual representation of a series of events in time. The use of timelines dates back to 17th century [32] when Joseph Priestly designed a visualization that showed the rise and fall of empires in Europe's history. In the modern era, timelines have become prevalent in our daily lives as the de facto representation to show financial trends, weather details, and meeting schedules. Timelines are most commonly drawn linearly [6], where the events are organized along a straight line. In practice, however, we can find abundant examples of timelines where events are arranged in non-linear shapes like circles, spirals, grids, and other arbitrary arrangements [6].

The visualization literature provides sufficient evidence that the layout and orientation of visualizations affect user's analytical task performance [14, 18]. In a classic study, Cleveland and McGill [14] and later Heer et al. [18] studied how people decode data presented to them using different shapes and orientation. Their results show that visual representation affects human ability to accurately read the data. Extrapolating from previous research in Information Visualization, we argue that there might be different perceptual effects of representing temporal event sequence data with varying timeline shapes. However, existing work in timeline visualization evaluation has not measured the impact of timeline shape alone on user task performance for general temporal event sequence data.

Timelines represent temporal event sequence data. In a review of related work and discussion with experts working with such data in the fields of history and personal health informatics, we identified 3 types of temporal event sequence data: (1) A non-recurring series of events, e.g., world history where events do not repeat. (2) A recurring sequence of events where the events happen at specific intervals, e.g., a company's quarterly financial reports. (3) A mixture of recurring and non-recurring sequences. Our lives are the best example of the third category, where certain events are recurring like time of meals, whereas other events are not like the birth of a child.

Intuitively, we can reason that a circular timeline may help readers notice a repeating pattern in the data, while a non-recurrent type of dataset may be best represented on a line to emphasize the linearity. However, there are no existing studies which systematically enquire whether the timeline shape effects reading time and accuracy for different types of temporal event sequence data.

Analytical tasks also play a role in a user's ability to read visualizations [10, 28, 33]. We interviewed two experts to create a list of common tasks on temporal event sequence data. We found that in a general day-to-day setting the experts used timelines for 4 analytical tasks: (1) a When task to identify the time associated with an event, (2) a What task to identify an event associated with a time, (3) a Find task to spot a data point on timeline based on both time and event, and (4) a Compare task which requires finding the distance between events.

In this paper, we present the first study which evaluates the readability of timeline shape alone on user task performance for general temporal event sequence data. In a crowd-sourced experiment, we compare 4 timeline shapes — horizontal line, vertical line, circle, and spiral — using 3 types of temporal data — recurrent, non-recurrent, and mixed. Our study is carefully designed to evaluate timeline shapes using common everyday tasks with familiar-looking datasets. E.g., find the date associated with an historical event on a timeline or lookup your daily schedule to find what are you supposed to do tonight at 8pm. In a within-subjects study design, we measured time to complete a visualization task and the task accuracy across the 4 timeline shapes.

We found evidence that task completion time is dependent on the choice of the timeline shape. Specifically, linear shapes were on an average faster to read. Our quantitative results are backed by qualitative feedback, where we see a strong preference for linear shapes among the participants.

Contributions: We contribute an overview of common timeline visualization tasks. We use these tasks to design a crowd-sourced experiment which compares readability across 4 common timeline shapes — horizontal line, vertical line, circle, and spiral. Based on our results, we recommend timeline design considerations for people interested in visually presenting temporal event sequence data.

RELATED WORK

Visualization of temporal event sequence data has a large body of literature. Existing work lays a strong emphasis on applied research of timeline visualizations and creating novel ways to visualize temporal event sequence data. In our review of related work, we summarize the research on timeline shape and how shape is used in practice. Unlike the design of timelines, evaluation of timelines is still in the early stages with only a few empirical studies evaluating the design of timeline shapes. In this section, we also discuss existing research evaluating timelines and identify areas of knowledge gap.

Timeline Visualizations and Tasks. Timeline visualizations come in a variety of shapes and are designed with a multitude of encodings and data. Brehmer et al. [6] present a design space for timelines used in storytelling. This design space identifies 5 representations (shapes) timelines take: linear, radial (circle), grid, spiral, and arbitrary. Aigner et al. [1] present a comprehensive book covering the visualization of time-oriented data. Much of the material covered is directly related or relevant to timeline visualization. They identify two possible arrangements of data: linear and cyclical.

In addition, Aigner et al. [1] include a review of several timeline visualizations with varying shapes, data arrangements, and encodings. These include examples of linear [2, 27], circular [13, 20], and spiral [11, 36] timeline visualizations. In each of these cases the timeline shape was chosen to best emphasize certain features of the data or to assist specific visualization tasks. E.g., linear timeline visualizations are generally used for navigation of the data or to display a linear sequence of events. However, circle and spiral timeline visualizations are used to highlight the cyclical or serial-periodic nature of data — such as the seasons of the year. Aigner et al. [1] also characterize several user tasks that are commonly performed with temporal data. These tasks included finding at what time events happened, finding what events happened at a specific time, comparing the interval of time between two events, identifying groups of related events, and more.

Timeline Evaluation. Previous work has evaluated the utility of various timeline visualizations for specific tasks. Schwab et al. [30] evaluated how quickly users can navigate to a particular data point on a timeline using pan and zoom techniques.

Brehmer et al. [7] compared linear vs. circular layout timelines on mobile devices on their efficacy for showing ranges. Recently, Waldner et al. [35] published a study on timeline shape focused on comparing juxtaposed radial charts vs. horizontal linear bar charts and two juxtaposed 12-hr charts vs. a single 24-hr chart. Their design intrinsically measured the combination of shape and juxtaposition technique. Our work complements Waldner et al.'s by evaluating the effect of simple timeline shapes alone and for additional shapes. Moreover, the data used by Waldner et al. was limited to a 24-hour setting while we test events more generally over different time scales. Furthermore, they used quantitative data associated with the occurrence/frequency of events while we focus on events only so as to measure effects for different types of event sequences.

Apart from empirical evaluations, researchers have evaluated the usability of various timeline shapes. Larsen et al. [22] found in a qualitative user study that spiral timelines were effective in identifying cycles in the data. Separately, Nguyen et al. [25] evaluated a novel linear timeline layout and found the layout to be easy to learn and effective for sensemaking.

RESEARCH QUESTION AND HYPOTHESES

Our primary motivation is to study how different timeline shapes affect a user's ability to read data with a timeline visualization. To answer this research question, we measure a user's ability to perform a visualization task in terms of their accuracy and completion times. Further, we also take into account their overall preferences when they are performing visualization tasks. We present the formal questions which compare these measures, as the following hypotheses:

1. We will not observe a substantial difference in time or accuracy between timeline shapes in any dataset.
2. We will observe higher user preference and confidence with linear timelines for all datasets.

Based on our interviews with experts, literature review, and intuitions as visualization researchers, we expect the time difference between timeline shapes to be inconsequential for all tasks and questions. Timeline shape will not be detrimental to participants' ability to complete tasks in a timely manner. However, we expect users to show a preference towards linear vertical and horizontal timelines as these are the most conventional.

STUDY

The stimuli used in this study were carefully and purposefully designed, discussed, and refined. In this section, first we motivate and explain our stimuli design. After stimuli description, we present our experimental methods. The experiment design measures the effect of the independent timeline design variable, on the dependent accuracy and time variables for 4 visualization tasks. Here, we walk you step-by-step through the experiment design and procedure.

Stimuli

Datasets. After careful inspection of the literature related to timelines and their use [1, 6], we identified 3 major types of data that are represented using timelines: non-recurrent, recurrent, and a combination of both (mixed). A non-recurrent dataset has distinct entries and does not repeat itself whereas a recurrent dataset has entries that repeat after a certain interval. A mixed dataset has both these properties, i.e, it contains both distinct and repeating entries. We created a dataset for each of these characteristics using fictitious data. This was done to avoid the potential for participants to be biased by previous knowledge of real data. For example, a participant may already know the sequence of events in a 'history of Egypt' dataset. This would give them an advantage over other participants and potentially skew results. Fictitious data avoids this problem and ensures participants will have no prior knowledge of the event sequences. Our 3 datasets — history (non-recurrent), gardening (recurrent), and schedule (mixed) — are demonstrated in Table 1.

Datasets		
History: Non-recurrent	**Gardening: Recurrent**	**Schedule: Mixed**
912, Major epidemic	Spr '18, Dittany	Mon 8AM, Wake up
915, Population plummets	Sum '18, Gurdyroot	Mon 12PM, Lunch
917, Tribes migrate	Aut '18, Puffapod	Mon 4PM, Social Hour
918, They settle	Win '18, Wolfsbane	Mon 8PM, Gym
921, Farming starts	Spr '19, Dittany	Tue 8AM, Wake up
924, Agriculture improves	Sum '19, Gurdyroot	Tue 10AM, Reading Group
925, Bartering ends	Aut '19, Puffapod	Tue 12PM, Lunch
927, Temples built	Win '19, Wolfsbane	Tue 6PM, Movie Night
...	...	...

Table 1. A overview of the datasets used in the experiment. The colors except black indicate recurrent events. (a) A non-recurrent historical timeline with invented dates and corresponding events. (b) A recurrent planting schedule of 4 different invented plants. (c) A mixed dataset consisting of an invented schedule with both repeating and non-repeating events.

Overview of Visual Stimuli. We use the design space proposed by Brehmer et al. [6] to finalize our experimental stimuli. Brehmer et al. divide the timeline design space into 5 categories: Linear, Radial, Grid, Spiral, and Arbitrary. We evaluate 3 of the 5 shapes: Line, Radial (called Circle in this paper), and Spiral. We exclude Grid because it is not a line-based representation, and Arbitrary because it is difficult to assign a single shape given its multiple forms. We ultimately developed 4 timeline shapes: a horizontal line, a vertical line, a circle, and a spiral. Our timeline design aims to maximize the effect of timeline shape and reduce the impact of other visual embellishments. As a result, our timelines do not use any form of color-coding. Only ticks and textual labels are used to represent events. All the labels are horizontal. To support comparison across the shapes, we keep the font size and style consistent across the designs. The final timelines are black lines with ticks indicating the position of events. Each event is further represented by a label for the date and another one for the name of the event. Events on the timeline are positioned at distances proportional to their distance in time. Each timeline is drawn within a pre-determined frame size which is rendered using 70% of the user's screen size. The timelines were implemented in JavaScript using D3 [5].

User-Centered Timeline Design. A timeline design has concerns beyond its visual presentation. E.g., how do people read timelines? We did not want to make an assumption about the readability of the timeline based on our intuition. To understand how people read the 4 timeline shapes used in our experiment, we conducted an in-person survey with 11 partic-

Shape	Way to read	Starting point
−	Left to right	Left
|	Top down	Top
○	Clockwise	Top
◎	From the inside	Top, going to the right

Table 2. Results of timeline readability survey. Here we show the most common readability technique that emerged for each timeline shape.

ipants from Northeastern University, Boston. In the survey, we asked the participants how they will read each of the 4 timelines. Our findings, summarized in Table 2, guided our stimuli design for the full study.

We used a similar approach to find the maximum number of points that can be shown on each shape without making it look overwhelming or cluttered. We built a web application with sliders to adjust the font size and number of points while simultaneously showing the effect of the changes on each of the shapes. We then asked participants to adjust the sliders until they reached the maximum number of points that still felt comfortable to read. Based on our findings, we decided to display 12 data points on each one of the visualizations.

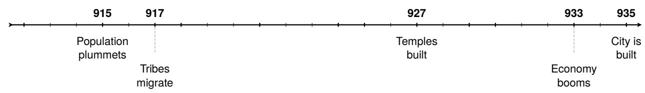
Final Stimuli Designs. We compare the readability of 4 timeline shapes with 3 temporal event sequence datasets. To test each shape with each dataset, we represent all 3 types of datasets on all 4 timeline designs. This led to 12 unique combinations of timeline shape and dataset. We present an overview of these shapes in Figure 1.

General Design Patterns. Key design elements were consistent across all 4 timeline shapes. Each timeline contains 12 data points and their dimensions are proportionally regular to account for varying screen sizes of participants. Each timeline has arrows denoting the beginning and end. Events are displayed with two labels: name of the event and date.

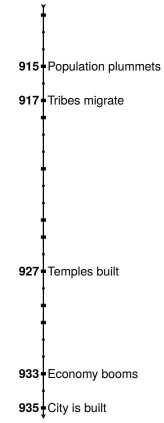
Dataset-Specific Design Patterns. There are some visual concerns related to the representation of each dataset with each shape. One of these is to aid readability between two labeled events. To support the users in tracking time difference between two labeled events, we use small ticks to display units of time between the events. The tick is placed according to the dataset: for the history dataset each tick denotes one year, for the gardening dataset each tick denotes one season, and for the schedule dataset each tick denotes two hours.

Horizontal and Vertical Line. Figure 2a shows the history timeline represented on a horizontal line − . We alternate the vertical positioning of the labels to avoid the text overlapping. The vertical line | history timeline is shown in Figure 2b. In both a difference in time is represented as the same Euclidean distance along the line.

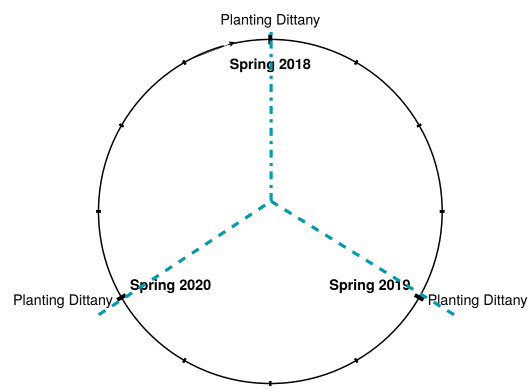
Circle and Spiral. On circle and spiral, a difference in time along the curve is represented by the angle difference from the center. For instance, if two events are separated by one year then the separation between them be an equal angle. In Figure 2c, we show how 3 equal-angle sectors preserve the distance between 3 recurrent events. In the spiral, preserving the



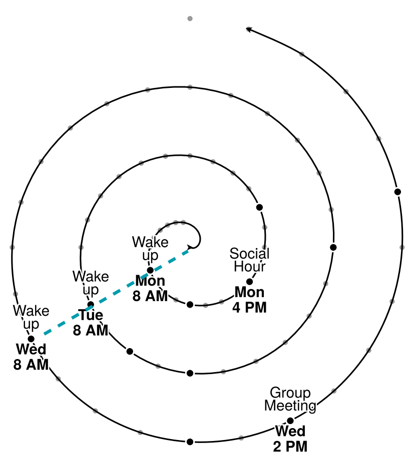
(a) A non-recurrent dataset on a horizontal line − timeline.



(b) A non-recurrent dataset on a vertical line | timeline.



(c) A recurrent dataset on a circle ○ timeline.



(d) A mixed dataset on a spiral ◎ timeline.

Figure 2. Examples of the stimuli used in the experiment. All the datasets have been simplified to aid readability within the paper.

relative angle from center aligns periodic activities along an axis. As shown in Figure 2dfor a mixed dataset, a recurring pattern 'Wake up' aligns along an axis. The final experimental stimuli are available at https://osf.io/qr5yu/

Experiment

Participants. We recruited 192 participants. The experiment was expected to last 5 minutes, and each participant received $2 base pay plus a bonus of $0.2 for each correct question after the sixth correct one. Our participants self-reported their demographics.. By gender 53% were male, 31% were female, while the rest chose not to answer. By education 32.8% had a Bachelor's degree, 17.7% had a high school degree, 16.6% attended some college but do not have a degree, and the rest had either a Master's degree, an Associate Degree, a Doctorate, had no degree, chose not to answer. By employment 62.5% were employed for wages, 17.2% were self-employed, 5.2% were out of work, and the rest were either students, homemakers, retired, unable to work, or chose not to answer.

Experiment Tasks. In order to determine the user tasks most relevant for timelines we interviewed a professor working in the history field and a PhD student with personal health data from Northeastern University, Boston. Both of them were selected because their research dealt with timelines and sequences of temporal information. The semi-structured interviews lasted about 1 hour. We asked them to discuss the information that is important to show on a timeline and what kind of tasks would be relevant for the use of timelines in their own field. We based our task analysis on the information we obtained from the interviews using Brehmer and Munzner's task taxonomy [9]. If we say that the target of a task is the name of an event, and that the year is the position, we can formulate specific questions that correspond to tasks. A locate task, for example, can be formulated on a timeline containing daily events as "At what time was lunch on Monday?" In this way, the question gives the target (lunch) and asks for the location (at what time?).

- **When** did <event> happen?
  Example: When did the earthquake happen?
  In this case, we are asking for the location of a target. The location is unknown but the target is known. Users would need to explore the timeline and report the year as an answer.
  Tasks: **Discover → Locate → Identify.**
- **What** happened at <time>?
  Example: What happened in 1999? What happens at the start of the timeline?
  Location is given, target is not. Users will look at the location and report the name of the event.
  Tasks: **Discover → Browse → Identify.**
- **Find** <event> that happened at <time>?
  Example: The earthquake happened in 1898. Click on it.
  Participants will answer this type of question by clicking on a specific point on the timeline. This question gives to the users both the target and the location.
  Tasks: **Discover → Lookup → Identify.**
- **Compare** time of <event1> relative <event2> or <event3>?
  Example: Did Cleopatra live closer to the launch of the first iPhone or the construction of the Pyramids?
  Users need to find 3 data points on the timelines and then compare the distance between pairs of data points.
  Tasks: **Discover → Explore → Compare.**

Experiment Design. Upon approval from the Institutional Review Board of Northeastern University, we recruited participants from MTurk. In order to be able to use the platform, we developed a web application to run the test directly on the participants' home computers. After running an initial pilot with 16 participants, we used a power analysis (see below) which indicated that we needed to recruit 192 participants. They were randomly but equally divided into 4 main groups, and each group of participants was assigned questions belonging to a particular task to be performed on a timeline (i.e., group 1 only performed the "when" task, group 2 only performed the "what" task, etc.). In order to minimize learning and ordering effects, we used Latin squares to determine the sequence of datasets and timeline shapes that the participants were shown.

Experiment Apparatus. MTurk was used to recruit and pay the participant, but the experiment was actually hosted on our own domain. It was served as a web app in which the participants could interact with both keyboard and mouse to go through the tutorial and answer the questions. All the code was written in JavaScript, ran on the participant's own device, and it rendered the stimuli and questions at run time in svg in the participant's browser using their screen size to determine the size of the visualization.

Experiment Procedure. After accepting the HIT, participants were asked to agree to our informed consent. They were then asked to complete a short tutorial that introduced them to the different timeline shapes used in the test and how to interact with the web application. The test required each participant to answer 12 questions. Groups 1, 2, and 4 answered multiple choice questions while group 3 was asked to click on timelines to find the answers. At the end, participants were prompted to answer an optional survey (described in the next section). After completing the study participants were given a unique code that they had to copy and paste into a prompt on MTurk so that they could get paid based on their performance.

Survey. The participants were given a short optional survey upon completion of the 12 questions. The survey contained questions about the demography of the participants, their feedback and whether they used any strategy while answering the questions. We also asked the 4 groups of participants which timeline shape was the most readable for each of the 3 different datasets. Readability here denotes which shape was the easiest to follow and read given a dataset. This was included to get an idea of which shapes worked best with what kind of dataset. The reason behind asking the "strategy" question was to make sure that the participants used the timeline shapes to navigate themselves in finding the correct answer and not some other strategy that depended less on the shape.

Data Analysis

To analyse the results of our experiment we used null hypothesis significance testing as well as interval estimation of effect size [15]. The analysis plan was pre-registered before we conducted the experiment and is available at https://osf.io/qr5yu/

Required Sample Size Based on Power Analysis. In order to effectively test for the effects of time and accuracy across the different timeline shapes, we conducted a power analysis to estimate the number of participants needed to identify a statistically significant difference with a Type I error rate α of 0.05. We used the G*Power [16] for power analysis. G*Power does not support Friedman's non-parametric repeated measures test to compare two or more samples. As an alternative, we estimated the sample size by comparing two samples using the one-tailed Wilcoxon signed rank test. Since we estimated the sample size pairwise we used conservative input parameters to handle the multiple comparisons and to estimate the upper bound on the number of participants required for the study. The parameters used in the study were Type I error rate α of 0.83% (Bonferroni corrected) at a power level of $1-\beta$ of 95%. The effect size was individually calculated using the mean completion time from the pilot study. The number of participants was estimated to be 48 for each task when rounded up to the nearest multiple of 12 to ensure that each group in the Latin square had the same number of participants. Thus a total of $48 \times 4 = 192$ participants were recruited.

Outlier Removal. Before proceeding with statistical tests comparing the time and accuracy distributions of different timeline shapes we identified a sizable set of data points that had particularly large measurements of time. Such outliers are quite common in time-based measurements from user studies and, if included without any special consideration, they can severely affect the quality of the statistical analysis. Outliers in the time data are usually not attributed to a group of participants that found a question excruciatingly difficult but most likely they were distracted during that question. In the proceeding analysis we considered a time data point for a given task as an outlier if its value is two or more standard deviations greater than the mean time for its given task. Across all 4 tasks we classified a total of 66 time data points as outliers and they originated from 47 participants. Those outlier data points were not part of any of our statistical analysis.

Time Analysis. In order to ensure that our statistical tests would not violate any of their necessary assumptions we examined the time distributions per dataset for each (task, timeline shape) pair. In particular, we tested for the normality of each time distribution using Q-Q plots and the Shapiro-Wilk test [31] and interpreted the p-values holistically rather than dichotomously. We determined that our time data was not normally distributed. We also performed Box-Cox transformations [29] for each time distribution and not all of them were transformable to a normal distribution using the same exponent, thus we considered *non-parametric statistical tests* that don't assume a normal distribution of the data.

In order to examine if there was a difference in the time distributions between the 4 timeline shapes for a given dataset we ran a Friedman test [17] comparing the 4 distributions with the null hypothesis that all timeline shapes have the same time distribution. A Friedman test is an appropriate choice as it is the *non-parametric equivalent* of a one-way ANOVA test with repeated measures. Notice that each participant in a given task saw — i.e., repeated — each timeline shape 3 times. If the null hypothesis of the Friedman test was rejected that implies that there was at least one pair of timeline shapes whose time distributions were different at the $\alpha = 5\%$ level. To identify which pairs of timeline shapes were different we ran a Nemenyi test [24] for each dataset as our post-hoc analysis. The Nemenyi test performs all pairwise comparisons between our different timeline shapes and the p-values it reports for each pair are adjusted for multiple hypotheses testing. We then interpreted these p-values more holistically in the context of our interval estimates (see below).

To examine if there is difference in time between the 4 timeline shapes for a given (task, dataset) pair we ran a Friedman test [17] for each pair with a Nemenyi test for the post-hoc analysis just like we did for our per-dataset analysis as we explained previously. Notice, because the Friedman test requires that each participant to have a data point in each distribution (i.e., timeline shape) we sometimes run into the problem of missing data due to the removal of outlier time data from our analysis. E.g., if a participant's time performance for the spiral timeline was classified as an outlier we cannot run the Friedman test using that participant's data even though we have their time performance on the other timeline shapes. As a result, workers whose data included an outlier were pruned from the Friedman test. It's important to emphasize however that the number of workers pruned for each group that the Friedman test was applied to was relatively small at around 0–15% of the total number of participants in the group being tested.

Accuracy Analysis. Because each participant sees only one question per (dataset, timeline shape) pair their accuracy for a given pair is either 100% (correct) or 0% (incorrect). As a result, our accuracy analysis resorts to *non-parametric statistical tests* due to the binary nature of our data distribution. In particular, we use a Chi-Square test for independence [23] for each (task, dataset) pair with the null hypothesis that there is no association between timeline shape and the response distribution. In other words, the null hypothesis is that user response (i.e., accuracy) is not dependent on the timeline shape. Just like our time analysis interpreted the p-values holistically rather than dichotomously.

Interval Estimation of Effect Size. Although null hypothesis significance tests are ubiquitously used to test for the validity of alternative hypotheses, they can also be problematic due to their dichotomous nature [4] and their imposed arbitrary p-value < .05 threshold that's used for denoting scientific findings of statistical significance. In addition to our statistical tests with null hypothesis significance testing, we also report effect sizes with interval estimates [15] in order to uncover more nuanced trends in our data that cannot be represented by just a p-value. To do so we provide the 95% confidence intervals (CIs) via bootstrapping[1] for each (task, dataset) pair to indicate the range of plausible values of the mean completion

[1]Bootstrapping involves randomly drawing observations from the experimental data with replacement in order to assess many alternate datasets, and thus use the variability across these datasets as a proxy for sampling error [15].

		When				What				Find				Compare			
		C	LH	LV	S	C	LH	LV	S	C	LH	LV	S	C	LH	LV	S
Mixed	C		0.59	0.90	0.90		0.39	0.00	0.90		0.06	0.01	0.83		0.90	0.20	0.90
	LH	0.59		0.64	0.90	0.39		0.05	0.63	0.06		0.90	0.32	0.90		0.47	0.90
	LV	0.90	0.64		0.90	0.00	0.05		0.00	0.01	0.90		0.12	0.20	0.47		0.24
	S	0.90	0.90	0.90		0.90	0.63	0.00		0.83	0.32	0.12		0.90	0.90	0.24	
Non-Recurrent	C		0.53	0.71	0.01		0.90	0.03	0.66		0.36	0.36	0.90		0.90	0.90	0.06
	LH	0.53		0.90	0.35	0.90		0.01	0.79	0.36		0.90	0.12	0.90		0.90	0.01
	LV	0.71	0.90		0.20	0.03	0.01		0.00	0.36	0.90		0.12	0.90	0.90		0.02
	S	0.01	0.35	0.20		0.66	0.79	0.00		0.90	0.12	0.12		0.06	0.01	0.02	
Recurrent	C		0.83	0.79	0.00		0.47	0.71	0.56		0.90	0.89	0.33		0.90	0.90	0.03
	LH	0.83		0.90	0.01	0.47		0.90	0.03	0.90		0.74	0.21	0.90		0.90	0.05
	LV	0.79	0.90		0.01	0.71	0.90		0.10	0.89	0.74		0.74	0.90	0.90		0.02
	S	0.00	0.01	0.01		0.56	0.03	0.10		0.33	0.21	0.74		0.03	0.05	0.02	

Figure 3. Exact *p*-values for the time analysis from the Nemenyi post-hoc test for the 4 tasks (columns) and 3 data types (rows). Low *p*-values indicating differences are highlighted in orange. Note that the matrix is symmetric.

time and mean proportion of correct responses [12]. We also examine the 95% CI for the mean per-worker log change in completion time [34]) using the the most well-known timeline shape (linear horizontal) as the comparison baseline. Similar examples of using estimation for this type of analysis are presented in [3, 8]. Note one exception to our pre-registration: we had planned to use the ratio of completion time instead of the log change in completion time but discovered that the former is asymmetric [34].

Qualitative Data Analysis. In order to get a sense of which timeline shapes did well in terms of readability across all the datasets in each the 4 tasks (When, Where, Find, and Compare), we simply calculated the total number of responses for readability for each category (horizontal line, vertical line, circle, and spiral) and found their percentages using the total number of responses per task. The results are discussed later in the paper. To analyze the comments left by the participants we used open coding [21]. In particular, we applied conventional content analysis where the categories of code are derived directly from the text data [19].

RESULTS

Quantitative Results. The results of our study are summarized in Figure 4, which shows the mean completion time, mean log change in completion time, and mean proportion correct for each combination of task, dataset, and timeline shape. Moreover, the results of pairwise Nemenyi tests, from our post-hoc analysis of the Friedman tests, are shown in Figure 3. We identified the following meaningful differences in time between pairs of timelines shapes for a given (task, dataset) pair. We use the timeline glyphs: ○ , — , | , ◎ to represent circle, linear horizontal, linear vertical, and spiral timeline shapes, respectively. The < symbol between timeline glyphs indicates that the mean time of the shape on the left is likely meaningfully smaller than the mean time of the shape on the right. The = symbol indicates likely similar mean times.

When Task.

- Mixed data: ○ = | = ◎ ($p = .90$ for all) and — = ◎ ($p = .90$).
- Non-recurrent data: ○ < ◎ ($p = .01$) and — = | ($p = .90$).
- Recurrent data: ○ < ◎ ($p < .001$), — < ◎ ($p = .01$), | < ◎ ($p = .01$), and — = | ($p = .90$).

What Task.

- Mixed data: | < ○ ($p = .01$), | < — ($p = .05$), | < ◎ ($p < .001$), and ○ = ◎ ($p = .90$).
- Non-recurrent data: | < ○ ($p = .03$), | < — ($p = .01$), | < ◎ ($p < .001$), and — = ○ ($p = .90$).
- Recurrent data: — < ◎ ($p = .03$) and — = | ($p = .90$).

Find Task.

- Mixed data: | < ○ ($p = .01$) and — = | ($p = .90$).
- Non-recurrent data: ○ = ◎ ($p = .90$) and — = | ($p = .90$).
- Recurrent data: ○ = — ($p = .90$) and ○ = | ($p = .89$).

Compare Task.

- Mixed data: ○ = — = ◎ ($p = .90$ for all).
- Non-recurrent data: — < ◎ ($p = .01$), | < ◎ ($p = .02$), and — = | = ○ ($p = .90$ for all).
- Recurrent data: ○ < ◎ ($p = .03$), — < ◎ ($p = .05$), | < ◎ ($p = .02$), and — = | = ○ ($p = .90$ for all).

Through our accuracy analysis we failed to find any meaningful differences between any pair of timeline shapes for any given (task, dataset) pair. Details at https://osf.io/qr5yu/

Qualitative Results. As part of the survey, participants were asked to select the timeline shape they perceived as the easiest to read for each dataset. Figure 5 shows the results. The horizontal line — was largely chosen as the easiest to read, especially for the non-recurrent dataset. Responses where more mixed for the recurrent dataset where, depending on the task, circle ○ and spiral ◎ were selected more.

We also received some interesting comments that pertained directly to the shape of the timelines. This gave us a notion of how the participants felt about certain shapes to an extent that they cared enough to express it in an optional feedback section. In this case too the majority of the comments were in favor of the horizontal and vertical lines. The comments were of the similar trend: *"...the vertical line is easy to read"*, *"The horizontal and vertical graphs were easiest"*, etc. A few remarked in favor of the circular shape: *"...but circles were easiest for my eyes"*. The spiral shape also had a few admirers under its belt with comments like: *"The planting dataset with the spiral is a good example of a timeline that is functional and visually appealing"*. However, on the flip side, some of these participants revealed concerns about the spiral shape, stating it to be difficult and confusing. Their comments echoed similarly: *"I found the spiral hard to work with"*, *"...the spirals were confusing"*, *"The spiral one was really hard"*.

DISCUSSION

In this paper, we present a novel study comparing timeline shapes. Our participants completed a controlled visualization experiment and we measured their performance in terms of time and accuracy for completing analytical tasks. Although we hypothesized that timeline shapes would not affect a person's performance, our results suggest the opposite, and we found timeline shape does indeed affect how people read a timeline visualization. In this section, we discuss the relevance of our experimental design and quantitative (Accuracy and Time) and qualitative user performance measures. After establishing the significance of our experimental design, we analyze the results and discuss the implications of findings on the design principles for timelines.

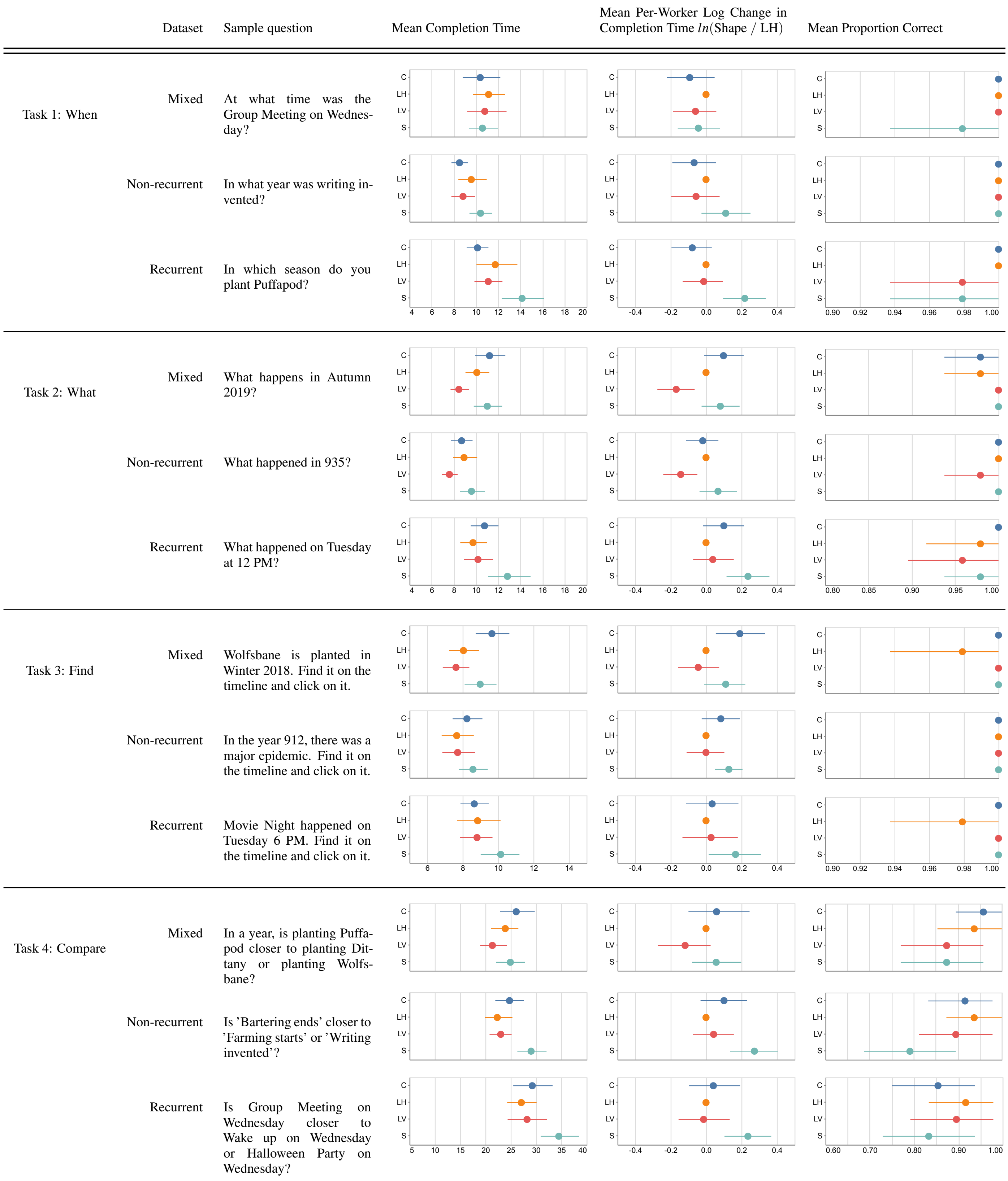


Figure 4. Quantitative experiment results for the 4 experimental tasks, each with 3 datasets and 4 timeline shapes — circle (C, ●), linear horizontal (LH, ●), linear vertical (LV, ●), and spiral (S, ●). For reference, we show a sample question for each task-dataset pair. Error bars show the bootstrapped 95% confidence interval (CI) of the mean [12, 15]. Left: Mean completion time. Middle: Mean per-worker log change in completion time using each worker's linear horizontal (LH) time as the baseline. Note that natural log ratios are the only symmetric, additive, and normed indicators of relative change [34]. Right: Mean proportion of correct responses. In the left and right columns the horizontal axis scale is relative for each task.

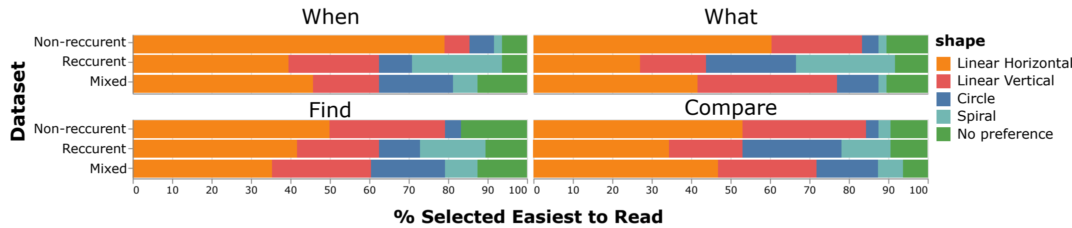


Figure 5. Comparison between the responses of the users when asked which shape was the easiest to read for each one of the tasks.

Reflection on the Study Design. The design space of timeline shape and the ways to interact with them has grown rapidly over the last few decades [6, 30]. However, evaluation of this design space is still in the early stages (see Related Work). In this experiment, we evaluate the primary design factor of timelines, their shape. To evaluate the shape, we try to minimize all other factors to get the best estimation of the effect of timeline shape. Therefore, we chose a study design where the timelines were non-interactive with consistent design aspects such as the display aspect ratio, the font-size of tick labels, etc. as discussed in the Stimuli section. Our evaluation of minimalist timelines raises a question of the applicability of these results to timelines designed with complicated interactions or higher data density. We argue that a minimalist design also minimizes the chances of eliciting a difference in user's performance in the study. In other words, in a simple experiment, the error rate and total time to complete the task should be similar across the timeline designs. Therefore, if we notice any differences with a minimalist design, then the chances of noticing the differences should increase with the complexity of design and scale of data, unless the design is prepared to handle the scale of data specifically. The same argument can be made for the simplicity of the chosen tasks: many taxonomies of low-level tasks, e.g., the one we use [9], argue that sensemaking is built upon combinations of such low-level tasks.

Visualization tasks may require a varying level of cognitive effort. E.g., a task where a user has to find the time of an event (When task) is cognitively easier than a task to compare times for two events (Compare task). The dual-process of decision making [26] suggests that hard cognitive tasks may be slower to perform as they require more contemplative decision making. To reduce variation in task performance introduced by task complexity we use clear compartmentalization of tasks. Our study measures the difference in timeline readability with different timeline shapes. We argue that the total time and accuracy of a user with a timeline gives evidence about its readability. Quantitative measures do not provide a holistic view of the user perception of the readability of the timeline design and user perception and sentiment are important to understand. Consequently, we survey the perceived readability of a visualization design at the end of the experiment. We use both the quantitative and qualitative measures to predict the overall efficacy of timeline shapes for representing temporal event data.

Effect of Shape on Timeline Readability. In general, we found that participants are faster at reading information from linear timelines vs. circles and spirals. In 6 out of 12 comparison conditions, shown in Figure 4, either one or both of the linear timelines were meaningfully faster than the circle or spiral timelines. We also found linear shapes were as perceived more readable. A valid explanation for these results can be our participants' familiarity with linear timeline designs. In their survey of timeline visualizations Brehmer et al. [6] found 73% of timelines surveyed used a linear representation. Familiarity with the timeline design may also imply that participants had prior training to read the linear timelines or were faster to adapt to the linear design.

We found substantial evidence that participants were slower at reading the spiral timeline and also perceived it as less readable. In 7 out of 12 conditions, shown in Figure 4, spiral timelines were the slowest or on par with the slowest. These results are surprising, as prior research [11] presented evidence that spiral timelines work particularly well for representing serial-periodic data. We attribute the unfavorable results for spiral timelines to its lack of familiarity. Lack of familiarity with the timeline shape can make even a simple visualization task cognitively hard and lead to slow task completion times.

We found strong evidence that timeline shape affects task completion time. It is important to note that the time differences between timeline shapes were small in absolute values mean time differences ranging from 1–7 seconds (see Figure 4). However, this equated to a mean log change in completion time ranging $\pm$ 20 L% from the mean time for a linear horizontal timeline. We believe the main reason for this meaningful but relatively small effect size is the overall experiment design. We designed straightforward experiment stimuli, scaled the datasets to make the stimuli less overwhelming, and used common visualization tasks. As a result, we argue our experiment was not very difficult to complete for the participants. Some participants also acknowledged this in their feedback: *"All of the timelines were quite easy for follow and understand"*, *"The timelines were all very easy to read no strategy required"*. As a result we observe relatively small effects for time differences. However, posit that these result trends should hold generally and the effect size will increase as the task becomes harder.

Our evidence suggests timeline shape does not meaningfully affect user accuracy (see Figure 4). No general patterns seem to emerge from the results for the When, What, and Find tasks (see Figure 4). However, we can not conclude that the mean accuracy is completely similar either. As with the time effect size, our straightforward experiment design may be the reason for these accuracy results. In the compare task (see Figure 4), we notice a tendency for people to make more errors with spiral timelines. However, the results are not different enough to make any definitive claims.

Effect of Task on Timeline Readability. We never intended to compare the results across the different tasks. However, the difference in completion times and accuracy among the tasks may have real-world design implications. We found good evidence that participants were slower and less accurate with all the timeline shapes in the Compare Task of the study (see Figure 4). We argue this happened because the Compare task is cognitively harder than the other tasks. Furthermore, spirals may be cognitively harder to process, and using them for comparison tasks may compound task difficulty. As a result, we suggest designers use a simple timeline design like the linear design when the task is complicated, unless the task specifically requires the use of spiral timelines.

Effect of Dataset on Timeline Readability. In this study, we compare the 4 timeline shapes across 3 datasets. We do this to measure the association between timeline shape and dataset type. However, in our study we did not find any association between timeline shape and datasets. We thought the linear timelines would be more readable with linear data, while circle and spiral timelines will be easier to read with recurrent and mixed datasets. We did not find enough evidence to support this claim. Based on these results, designers have the flexibility of representing linear data on spiral timelines to make the visual design enjoyable — with the caveat of the expressiveness criteria discussed in Design Recommendations below. Vice versa, a complex mixed dataset can be represented with a linear timeline to aid faster readability.

Usability of Non-Linear Timelines: Circle and Spiral. We found that linear timelines are faster to read. However, usefulness and timeline readability are two different measures. There might be situations where non-linear timelines may be more informative, metaphorically similar to the dataset, or simply pleasing to eyes. In that scenario, the designers may choose aesthetic appeal over optimal speed. An important readability measure that goes in favor of non-linear timelines is the lack of difference in accuracy for cognitively easy tasks. If the consideration for readability is just accuracy, then we suggest liberal use of the timeline shapes — again with the caveat of the expressiveness criteria discussed in Design Recommendations below.

DESIGN RECOMMENDATIONS

Here we present a list of design recommendations derived from our results. We chose these recommendations based on their broad applicability. Our design recommendations are targeted at suggesting users the right design if their primary concern is to increase the readability of temporal event sequence data. However, based on domain and context, these recommendations may not directly be applicable. In such a case, we request designers to practice caution while using these recommendations.

Context, Usability, and Expressiveness. First of all, the importance of differences in timing are dependent on the context. E.g., in emergency medicine a difference of a few seconds may be much more relevant than for consuming a timeline published in a magazine. It is also important to consider the expressiveness of the data — how well the timeline shape can represent the underlying data. For instance, representing linear data on a circular shape may be misleading to the readers by inducing them into thinking there are recurrences in the data when there are none. We therefore suggest designers evaluate our recommendations based on their domain goals and specific use case.

Timeline Readability. Based on our results we formulate the following recommendations:

1. Use linear vertical timelines for situations which require fast data lookup.
2. Avoid spiral timelines when the task requires fast lookup.
3. If you use a spiral timeline also include a tutorial or visual cues to assist the user in learning and understanding.

Task Performance. Tasks which require long-term memory dependence like the Compare task are seem slower and less accurate on all the timeline designs. This effect is amplified with spiral timelines . We recommend using linear timelines for difficult tasks which require complicated decision making.

Dataset Flexibility. Within this study, we did not find that dataset choice affects the readability of the timeline. Therefore, we recommend designers to be flexible with their choice of timeline shape to maximize readability or improve engagement. However, if the dataset is complex, even for mixed data, we recommend using a linear timeline .

LIMITATIONS AND FUTURE WORK

In this study we measure the effect of shape on timeline readability. Besides shape, several other factors influence how people read timelines. Factors like data density, the use of interaction, visual embellishments like pictures on the timeline, and the difference in path length between a shape and another may also affect timeline readability. Future work can study the additional design characteristics proposed by Brehmer et al. [6] like scale (log or relative) as well as additional layouts (e.g., multiple stacked lines). The timeline design space is vast and we believe there is extensive room for future work to help us understand timelines better.

CONCLUSION

Timeline visualizations are pervasive in our everyday life. However, we know little about ways to design timelines effectively. We present a novel study to measure the effect of 4 common timeline shapes on timeline readability: a horizontal line , a vertical line , a circle , and a spiral . From our results, we learned that timeline shape affects the readability of timelines. More specifically, we found good evidence that the linear shapes support reading the timelines more quickly than the non-linear shapes. We also found evidence that non-linear spiral shape is not only perceived slowest by users but also leads to slower lookup of events with timelines. Future studies and real-word timeline designers should carefully analyze their data representations in light of our findings.

ACKNOWLEDGEMENTS

We thank the National Science Foundation for support, in part, under CRII award no. 1755901 and our reviewers & colleagues for their advice.

REFERENCES

[1] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. 2011. *Visualization of Time-Oriented Data*. Springer. DOI: http://dx.doi.org/10.1007/978-0-85729-079-3

[2] Paul André, Max L. Wilson, Alistair Russell, Daniel A. Smith, Alisdair Owens, and M.C. Schraefel. 2007. Continuum: Designing Timelines for Hierarchies, Relationships and Scale. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. 101–110. DOI: http://dx.doi.org/10.1145/1294211.1294229

[3] Lonni Besançon. 2017. *An interaction Continuum for 3D Dataset Visualization*. Ph.D. Dissertation. Paris Saclay.

[4] Lonni Besançon and Pierre Dragicevic. 2019. The Continued Prevalence of Dichotomous Inferences at CHI. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems (CHI EA '19)*. Article alt14, 11 pages. DOI: http://dx.doi.org/10.1145/3290607.3310432

[5] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3 data-driven documents. *IEEE transactions on visualization and computer graphics* 17, 12 (2011), 2301–2309. DOI: http://dx.doi.org/10.1109/TVCG.2011.185

[6] M. Brehmer, B. Lee, B. Bach, N. H. Riche, and T. Munzner. 2017. Timelines Revisited: A Design Space and Considerations for Expressive Storytelling. *IEEE Transactions on Visualization and Computer Graphics* 23, 9 (2017), 2151–2164. DOI: http://dx.doi.org/10.1109/TVCG.2016.2614803

[7] M. Brehmer, B. Lee, P. Isenberg, and E. K. Choe. 2019. Visualizing Ranges over Time on Mobile Phones: A Task-Based Crowdsourced Evaluation. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 619–629. DOI: http://dx.doi.org/10.1109/TVCG.2018.2865234

[8] M. Brehmer, B. Lee, P. Isenberg, and E. K. Choe. 2020. A Comparative Evaluation of Animation and Small Multiples for Trend Visualization on Mobile Phones. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 364–374. DOI: http://dx.doi.org/10.1109/TVCG.2019.2934397

[9] Matthew Brehmer and Tamara Munzner. 2013. A multi-level typology of abstract visualization tasks. *IEEE transactions on visualization and computer graphics* 19, 12 (2013), 2376–2385. DOI: http://dx.doi.org/10.1109/TVCG.2013.124

[10] Michael Burch, Natalia Konevtsova, Julian Heinrich, Markus Hoeferlin, and Daniel Weiskopf. 2011. Evaluation of Traditional, Orthogonal, and Radial Tree Diagrams by an Eye Tracking Study. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2440–2448. DOI: http://dx.doi.org/10.1109/TVCG.2011.193

[11] John V. Carlis and Joseph A. Konstan. 1998. Interactive visualization of serial periodic data. In *ACM Symposium on User Interface Software and Technology*. 29–38.

[12] James Carpenter and John Bithell. 2000. Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. *Statistics in Medicine* 19, 9 (2000), 1141–1164. DOI: http://dx.doi.org/10.1002/(SICI)1097-0258(20000515)19:9<1141::AID-SIM479>3.0.CO;2-F

[13] M. C. Chuah. 1998. Dynamic aggregation with circular visual designs. In *Proceedings IEEE Symposium on Information Visualization (Cat. No.98TB100258)*. 35–43. DOI: http://dx.doi.org/10.1109/INFVIS.1998.729557

[14] William S. Cleveland and Robert McGill. 1984. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *J. Amer. Statist. Assoc.* 79, 387 (1984), 531–554. DOI: http://dx.doi.org/10.2307/2288400

[15] Pierre Dragicevic. 2016. Fair Statistical Communication in HCI. In *Modern Statistical Methods for HCI*. 291 – 330. DOI: http://dx.doi.org/10.1007/978-3-319-26633-6_13

[16] Franz Faul, Edgar Erdfelder, Albert-Georg Lang, and Axel Buchner. 2007. G*Power 3: A flexible statistical power analysis program for the social, behavior, and biomedical sciences. *Behavior research methods* 39 (2007), 175–91. DOI: http://dx.doi.org/10.3758/BF03193146

[17] Milton Friedman. 1937. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Amer. Statist. Assoc.* 32, 200 (1937), 675–701. DOI: http://dx.doi.org/10.2307/2279372

[18] Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. 203–212. DOI: http://dx.doi.org/10.1145/1753326.1753357

[19] Hsiu-Fang Hsieh and Sarah E. Shannon. 2005. Three Approaches to Qualitative Content Analysis. *Qualitative Health Research* 15, 9 (2005), 1277–1288. DOI: http://dx.doi.org/10.1177/1049732305276687 PMID: 16204405.

[20] Daniel A. Keim, Jörn Schneidewind, and Mike Sips. 2004. CircleView: A New Approach for Visualizing Time-related Multidimensional Data Sets. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. 179–182. DOI: http://dx.doi.org/10.1145/989863.989891

[21] Shahedul Huq Khandkar. 2009. Open coding. *University of Calgary* 23 (2009), 2009.

[22] Jakob Eg Larsen, Andrea Cuttone, and Sune Lehmann Jørgensen. 2013. QS Spiral: Visualizing periodic quantified self data. In *CHI 2013 Workshop on Personal Informatics in the Wild: Hacking Habits for Health & Happiness*. 4.

[23] D. C. Montgomery and G. C. Runger. 2003. *Applied Statistics and Probability for Engineers*. John Wiley and Sons.

[24] Peter Nemenyi. 1962. Distribution-free multiple comparisons. In *Biometrics*, Vol. 18. 263.

[25] P. H. Nguyen, K. Xu, R. Walker, and B. L. W. Wong. 2014. SchemaLine: Timeline Visualization for Sensemaking. In *2014 18th International Conference on Information Visualisation*. 225–233. DOI: http://dx.doi.org/10.1109/IV.2014.14

[26] Lace M. Padilla, Sarah H. Creem-Regehr, Mary Hegarty, and Jeanine K. Stefanucci. 2018. Decision making with visualizations: a cognitive framework across disciplines. *Cognitive Research: Principles and Implications* 3, 1 (2018), 29. DOI: http://dx.doi.org/10.1186/s41235-018-0120-9

[27] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. 1996. LifeLines: Visualizing Personal Histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '96)*. 221–227. DOI: http://dx.doi.org/10.1145/238386.238493

[28] Peter Rodgers, Gem Stapleton, Bilal Alsallakh, Luana Michallef, Rob Baker, and Simon Thompson. 2016. A task-based evaluation of combined set and network visualization. *Information Sciences* 367-368 (2016), 58 – 79. DOI:http://dx.doi.org/10.1016/j.ins.2016.05.045

[29] Remi Sakia. 1992. The Box-Cox Transformation Technique: A Review. *The Statistician* 41 (1992). DOI: http://dx.doi.org/10.2307/2348250

[30] Michail Schwab, Sicheng Hao, Olga Vitek, James Tompkin, Jeff Huang, and Michelle A. Borkin. 2019. Evaluating Pan and Zoom Timelines and Sliders. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Article 556, 12 pages. DOI:http://dx.doi.org/10.1145/3290605.3300786

[31] S. S. Shapiro and M. B. Wilk. 1965. An Analysis of Variance Test for Normality (Complete Samples). *Biometrika* 52, 3/4 (1965), 591–611. http://www.jstor.org/stable/2333709

[32] Arthur Sheps. 1999. Joseph Priestley's time charts: the use and teaching of history by rational dissent in late eighteenth-century England. *Lumen: Selected Proceedings from the Canadian Society for Eighteenth-Century Studies/Lumen: Travaux choisis de la Société canadienne d'étude du dix-huitième siècle* 18 (1999), 135–154. DOI: http://dx.doi.org/10.7202/1012372ar

[33] John Stasko, Richard Catrambone, Mark Guzdial, and Kevin McDonald. 2000. An Evaluation of Space-filling Information Visualizations for Depicting Hierarchical Structures. *Int. J. Hum.-Comput. Stud.* 53, 5 (2000), 663–694. DOI: http://dx.doi.org/10.1006/ijhc.2000.0420

[34] Leo Törnqvist, Pentti Vartia, and Yrjö O Vartia. 1985. How should relative changes be measured? *The American Statistician* 39, 1 (1985), 43–46. DOI: http://dx.doi.org/10.2307/2683905

[35] M. Waldner, A. Diehl, D. Gračanin, R. Splechtna, C. Delrieux, and K. Matković. 2020. A Comparison of Radial and Linear Charts for Visualizing Daily Patterns. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 1033–1042. DOI: http://dx.doi.org/10.1109/TVCG.2019.2934784

[36] M. Weber, M. Alexa, and W. Muller. 2001. Visualizing time-series on spirals. In *IEEE Symposium on Information Visualization, 2001. INFOVIS 2001*. 7–13. DOI:http://dx.doi.org/10.1109/INFVIS.2001.963273

© 2019 IEEE. This is the author's version of the article that has been published in IEEE Transactions on Visualization and Computer Graphics. The final version of this record is available at: 10.1109/TVCG.2019.2934402

arXiv:1907.12663v2 [cs.HC] 15 Oct 2019

CerebroVis: Designing an Abstract yet Spatially Contextualized Cerebral Artery Network Visualization

Aditeya Pandey, Harsh Shukla, Geoffrey S. Young, Lei Qin, Amir A. Zamani, Liangge Hsu, Raymond Huang, Cody Dunne, and Michelle A. Borkin

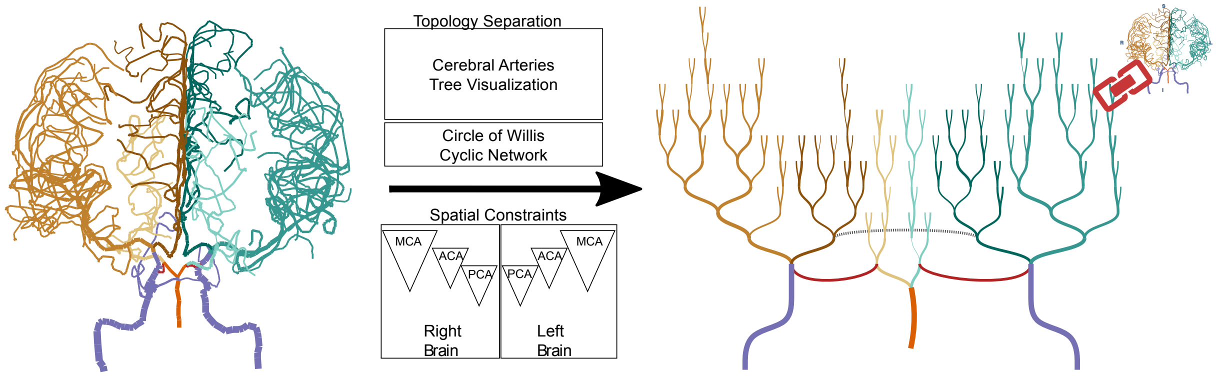


Fig. 1: CerebroVis is a novel network visualization for cerebral arteries. CerebroVis uses an abstract topology-preserving visual design which is put in spatial context by enforcing constraints on the network layout. Here we show the conversion of an almost symmetrical healthy human brain cerebral artery network from a 2D isosurface visualization (left) to CerebroVis (right). Each artery has the same categorical color in both views (see Sec. 3 for a legend).

Abstract—Blood circulation in the human brain is supplied through a network of cerebral arteries. If a clinician suspects a patient has a stroke or other cerebrovascular condition, they order imaging tests. Neuroradiologists visually search the resulting scans for abnormalities. Their visual search tasks correspond to the abstract network analysis tasks of browsing and path following. To assist neuroradiologists in identifying cerebral artery abnormalities, we designed CerebroVis, a novel abstract—yet spatially contextualized—cerebral artery network visualization. In this design study, we contribute a novel framing and definition of the cerebral artery system in terms of network theory and characterize neuroradiologist domain goals as abstract visualization and network analysis tasks. Through an iterative, user-centered design process we developed an abstract network layout technique which incorporates cerebral artery spatial context. The abstract visualization enables increased domain task performance over 3D geometry representations, while including spatial context helps preserve the user's mental map of the underlying geometry. We provide open source implementations of our network layout technique and prototype cerebral artery visualization tool. We demonstrate the robustness of our technique by successfully laying out 61 open source brain scans. We evaluate the effectiveness of our layout through a mixed methods study with three neuroradiologists. In a formative controlled experiment our study participants used CerebroVis and a conventional 3D visualization to examine real cerebral artery imaging data to identify a simulated intracranial artery stenosis. Participants were more accurate at identifying stenoses using CerebroVis (absolute risk difference 13%). A free copy of this paper, the evaluation stimuli and data, and source code are available at osf.io/e5sxt.

Index Terms—Network Visualization, Spatial Context, Abstract Design, Flow Network, Medical Imaging, Cerebral Arteries.

1 Introduction

Arteries in the human brain form a network of blood flow, and a blockage or leakage in this network can lead to life-threatening cerebrovascular conditions such as a stroke or aneurysm. Strokes alone are the fifth leading cause of death as well as a leading cause of serious long-term disability in the United States, and are globally the second leading cause of death after heart disease [35]. Early detection and diagnosis of these conditions is essential for effective life-saving treatment. Conventional diagnostics rely on an expert neuroradiologist identifying vascular abnormalities through examination of medical images (e.g., CTA, MRA). This data is commonly rendered in 3D to assist the doctor with identification of the abnormalities. However, prior research indicates that existing representations of the 3D cerebral arteries—e.g., isosurface, volume rendering, and Maximum Intensity Projection (MIPS)—introduce visual artifacts and task performance challenges such as overplotting/occlusion [21], false impression of geometry [21], and excessive artery bends.

In this design study, we present a novel 2D visualization of the cerebral artery system designed to assist doctors in the identification of cerebrovascular abnormalities. Inspired by existing visualization research which has demonstrated the effectiveness of 2D representations for spatial search tasks in other medical imaging cases, e.g., cardiovascular arteries [6] and connectomics [36], we present a novel 2D abstract representation of the cerebral arteries. To our knowledge, this is the first attempt to approach the cerebrovascular diagnostics tasks faced by neuroradiologists from the perspective of network science and using an abstract 2D visual encoding.

In this paper, we first offer a novel framing of cerebral arteries using network theory. Next, we characterize the domain goals and present them as network analysis tasks. In an iterative user-centered design with neuroradiology collaborators, we developed an effective abstract representation of the human cerebral artery network to assist neuroradiologists

• Aditeya Pandey, Harsh Shukla, Cody Dunne, and Michelle Borkin are with Northeastern University. E-mail: {pandey.ad, shukla.h}@husky.neu.edu, {c.dunne, m.borkin}@northeastern.edu
• Geoffrey S. Young, Amir A. Zamani, Liangge Hsu, and Raymond Huang are with Brigham and Women's Hospital. E-mail: {gsyoung, azamani, lhsu1, ryhuang}@bwh.harvard.edu
• Lei Qin is with the Dana-Farber Cancer Institute. E-mail: lqin2@partners.org

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

in identifying abnormalities. We discovered that this new representation is most effective when spatial context is included to help users understand the novel representation. In order to meet domain goals and satisfy design requirements, we developed a topology preserving network layout for cerebral arteries with spatial constraints to aid expert understanding.

We evaluate our new layout and the accompanying CerebroVis prototype in two ways: (1) assessing the robustness of the technique by examining 61 healthy brain scans [67] and (2) a mixed methods study with three neuroradiologists which included semi-structured interviews and a controlled experiment simulating intracranial stenosis diagnosis. We found that our layout and implementation correctly visualizes all 61 brain scans, that neuroradiologists were more accurate at identifying stenosis with CerebroVis vs. a 3D visualization (absolute risk difference 13%), and that neuroradiologists thought CerebroVis was easy to understand and a useful addition to their diagnosis toolbox.

Contributions: The primary contribution of this design study is a novel 2D, abstract, yet spatially contextualized cerebral artery network visualization and its open-source implementation in CerebroVis. We also contribute a novel definition of the cerebral artery system in terms of network theory, as well as a network visualization task abstraction for cerebrovascular diagnostics. Our evaluations with three neuroradiologists demonstrate the validity of our approach and confirm improved task performance for identifying cerebrovascular abnormalities as compared to a 3D visualization. Finally, we reflect on the general aspects of our design paradigm for the benefit of the visualization community at large.

2 Domain Background

Neuroradiology is a sub-specialty of radiology, the diagnosis of injuries and diseases with medical imaging, which focuses on conditions of the brain, spine, head, and neck. In this work, we focus specifically on cerebrovascular diseases and related brain blood flow abnormalities. In these diseases, which include stroke, aneurysm, and malformed vasculature, interrupted blood flow can deprive the brain of oxygen—with potentially lethal consequences. In order to diagnose cerebrovascular diseases, a neuroradiologist visualizes imaging data to evaluate the arteries and identify physical abnormalities. Abnormalities include vessel narrowing (stenosis), abnormal widening (fusiform aneurysm), berry shaped protrusion from normal arteries (thrombosis), and absence of normal vessels (occlusion). Treatment for cerebrovascular diseases, sometimes with extreme time sensitivity, can include administering blood thinners or clotting agents to restore or stabilize blood flow and image-guided procedures to mechanically restore or alter artery structures.

A large number of specialized cerebrovascular imaging techniques exist [42]. The most common techniques are Compute Tomography Angiography (CTA) [2, 41] and Magnetic Resonance Angiography (MRA) [2, 41]. In this paper we focus on MRA imaging due to its high resolution and open data set availability. MRA uses magnetic radio-pulses for imaging with injected contrast dye to increase visibility and resolution of arteries. The advantages of MRA as compared to CTA include higher quality imaging of soft tissues and no radiation (x-ray) exposure to the patient. The disadvantages of MRA include the long acquisition time (typically close to an hour) and an unpleasant experience for patients with claustrophobia (patients must lie still in a narrow tube).

Fig. 2 shows the three main methods used by neuroradiologists to visualize and analyze the cerebrovascular arteries. First (A) we have a 2D image "slice" (orthogonal cut) through the raw imaging cube (typically in DICOM format). Next (B) MIPS (maximum intensity projection) [23, 63] is a 2D projection of the 3D brain arteries. However, the projection makes it difficult to delineate individual vessels [21]. Third (C) is 3D rendering [39, 63] which preserves all spatial information and—through interaction—makes it possible to delineate individual arteries [21]. A key motivation for the development of our new method (D) was to clearly present each artery individually but obviate the need for interactivity.

3 Network Model of the Cerebral Artery System

Here we present a novel definition of the cerebral artery system using network theory. As will be discussed in Sec. 7, this network framing provided the essential insight and analytic strategies necessary for creating an effective 2D representation.

3.1 Networks and Trees Defined

A *network* represents entities (*nodes*) and the relationships between them (*edges*). In a *directed* network, each edge has a source node and target and encodes a meaningful relationship direction. In an *undirected* network, conversely, edges denote a bi-directional relationship. A pair of reciprocal *parallel* directed edges between two nodes can be modeled instead using a single undirected edge. If a network contains both directed and undirected edges it is termed *mixed*. A *path* is an alternating sequence of distinct nodes and edges which connect two nodes. A network is *connected* if there exists at least one path between every pair of nodes. If a node is reachable from itself via a path, that path is termed a *cycle* and the network is *cyclic*. A *tree* is an undirected, connected, acyclic network. This means there exists only one path between any pair of nodes. A *directed tree* uses directed edges, but edge directionality is ignored when ensuring only one path exists between any pair of nodes. For hierarchical data a *root* is assigned for the top level and edges in this *rooted tree* model parent-child relationships. In a network, the *degree* of a node is the total number of edges connected to the node. Nodes in the tree with a degree of at least two are called *internal nodes*, while nodes with a degree of one are *leaf nodes*. A tree in which each node has at most two children is called a *binary tree*.

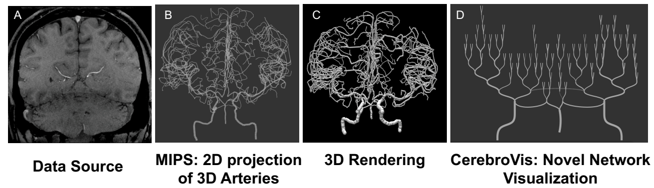


Fig. 2: Existing cerebral artery visualizations: (A) raw image "slices", (B) MIPS, and (C) 3D rendering. (D) Our CerebroVis visualization.

3.2 Cerebral Arteries as a Network

The human circulatory system can be modeled as a blood flow network with edges representing vessels with variable amount of blood flow. Arteries carry oxygenated blood away from the heart to the rest of the body including the brain. Veins return deoxygenated blood back to the heart via the lungs. In this paper we focus on arteries carrying blood to the brain. Our network of interest has three main components: (1) arteries carrying blood from the heart to the brain, (2) the flow regulating Circle of Willis, and (3) arteries distributing blood inside the brain. We discuss each of these below, including their diagnostic importance and a network theoretic data model. The categorical colors used throughout the paper and components of interest are illustrated in Fig. 3.

Arteries carrying blood to the brain: Four arteries supply blood to the brain: two *internal carotid arteries (IC)* and two *vertebral arteries*. The internal carotids provide blood to the *anterior* (front) part of the brain, with one serving the left hemisphere and the other serving the right. The vertebral arteries merge to form the *basilar artery (BA)* near the base of the brain. The basilar artery comes up the brain stem and supplies blood to the posterior (back) part of the brain. The carotid and basilar arteries end at the Circle of Willis.

Model: The ICs and the BA are each a continuous chain of directed, weighted edges where weight can denote width or blood flow.

Circle of Willis: The *Circle of Willis (CoW)* is part of the vasculature at the base of the brain and is important for blood distribution. The CoW regulates blood flow and provides redundant circulation—if part is blocked, blood can still flow to the brain. At this junction the incoming ICs and BA branch into six cerebral arteries (Fig. 3). The CoW connects the ICs with the BA through two *posterior communicating arteries (P. Comm.)*. It also connects the two anterior cerebral arteries (ACA) through a single artery called the *anterior communicating artery (A. Comm.)*. The communicating arteries allow flow from any input artery to the cerebral arteries and provide a failsafe in case of blockages.

Model: The Circle of Willis is an undirected, weighted cycle. Connected to the cycle are directed, weighted edges for input and output.

Distributing blood inside the brain: The *cerebral arteries* carry blood throughout the brain. Each major artery is named for the region of the brain it supplies. There are three pairs of cerebral arteries, with one

© 2019 IEEE. This is the author's version of the article that has been published in IEEE Transactions on Visualization and Computer Graphics. The final version of this record is available at: 10.1109/TVCG.2019.2934402

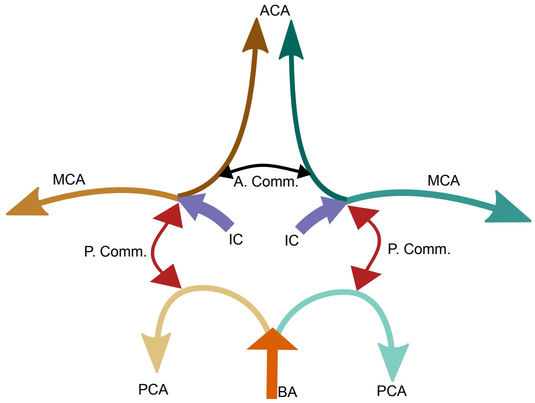


Fig. 3: Diagram of the bi-directional Circle of Willis (CoW) cycle in the cerebral artery network.

artery in each pair serving the left side of the brain and the other the right.

The *middle cerebral arteries (MCA)* branch from the internal carotid arteries (Fig. 3) and serve the middle of the brain. The *posterior cerebral arteries (PCA)* branch from the basilar artery (Fig. 3) and serve the back of the brain. The MCAs and PCAs have a directed blood flow stemming from the Circle of Willis. These arteries all split into two branches at each bifurcation point.

Model: The MCAs and PCAs are each directed binary trees.

The *anterior cerebral arteries (ACA)* also branch from the internal carotid arteries (Fig. 3) and serve the front of the brain. Similar to the MCAs and PCAs, they split into two branches at each bifurcation point and have directed flow from the Circle of Willis into the brain. However, there exists an undirected bridge between the left and right ACA: the *anterior communicating artery (A. Comm.)* (Fig. 3).

Model: The ACAs are directed binary trees joined by an undirected edge.

In Fig. 1 and Sec. 6 we illustrate and describe how this network definition helped us design our novel layout technique and visualization.

4 Related Work

4.1 Visualizing Cerebral Arteries and Brain Anatomy in 2D

As we introduced in Sec. 2, conventional 3D cerebrovascular visualizations suffer from issues of occlusion and clutter. Our goal is to develop a 2D visualization approach that can reduce clutter and assist in visual search tasks. Previous authors have focused on artery systems in other parts of the body, e.g., Borkin et al.'s heart visualization [6]. To our knowledge no prior work exists on visualizing cerebral arteries as a network. However, some authors have examined cerebral arteries using tree visualizations. E.g., to assist in detecting brain tumors Aydin et al. [3] used a tree visualization to represent the density of arteries in a region of the brain. Later Skwerer et al. [58] used a tree visualization for general analyses. However, prior work has not visualized the important cyclic Circle of Willis or its feeder arteries. Prior abstract tree visualizations also have not preserved any spatial position information about the arteries, while knowing exact and relative position is important for diagnosing some cerebral disorders.

Looking beyond arteries, previous authors have used 2D visualizations of other brain structures e.g., for connectomics: the study of neural connectivity in the brain. Abstract representations have been used extensively to visualize the functional and structural components of the brain (e.g., [1, 36, 45, 66]). Alper et al. [1] used adjacency matrix and node-link visualizations to study correlations between parts of the brain. They compared the accuracy of the two approaches and found the node-link visualization to be a more realistic representation of the brain. However, users faced problems with occlusion and performed worse than using a more abstract adjacency matrix. Likewise, in Neurolines [36] Al-Awami et al. point out the advantages of using an abstract subway map visualization for nanoscale neuron connectivity.

Take-away: 3D cerebral networks are structurally complicated but are important to understand in connectomics and neuroradiology. Abstract 2D network visualizations can provide valuable insights while avoiding issues of occlusion, but prior work on cerebral artery visualization has not included key anatomical structures and relative position.

4.2 Network Visualization

Readability & Aesthetics: The readability of node-link visualizations has been extensively studied, quantified, and summarized (e.g., [5, 17, 48, 59, 64]). Here we discuss three readability criteria which are particularly relevant for cerebral artery analysis: edge crossings, path continuity, and symmetry. Empirical research emphasizes the fact that a node-link visualization should try to minimize edge crossings [20, 48] and the idea has been broadly accepted by the community, with multiple algorithms that optimize for fewer edge crossings [11, 13, 19, 22, 60]. Path continuity is an important factor for the path following ability of users. Continuous curved paths are more easily perceived than polylines [37] and minimizing the number and angle of bends along the path improves path-finding task performance [37, 64]. Lipton et al. [43] concluded that a good network visualization displays as many symmetries as possible, and symmetry is especially important in our case as we want to enable comparison between the left and right halves of the brain.

Tree Layout and Comparison: While outside of the scope of this paper, an expansive survey of tree visualizations can be found on treevis.net [55]. Gomez et al. [27, 28] propose a design space of tree comparison techniques, of which we primarily focus on topology comparison. Previously Munzner et al. [47] compared the topology of two trees algorithmically. However, for comparing artery structures between the left and right brain we do not require exact one-to-one matching, nor would it be feasible. Instead, we use technique similar to that of Holten & van Wijk [33] which mirrors two tree visualizations in an orientation conducive for comparing hierarchical differences. There are two common categories of tree visualizations: space filling/implicit [54] techniques such as treemaps [57] and non-space filling techniques for laying out node-link visualizations such as Reingold-Tilford [50]. While space-filling visualizations are an effective way to optimize the usage of screen space to display node attribute values, they do not explicitly draw edges [54]. As discussed in the network data model (see Sec. 3), the human circulatory system is a blood flow network with edges representing vessels. Therefore, for artery sub-networks that have a tree structure we use node-link visualizations so as to provide edge marks on which we can directly encode blood flow.

Constraint-Based Layout: Constraint-based layouts impose restrictions on the resulting network spatialization, e.g., to reduce edge crossings [60], uncover structures or patterns which were not otherwise visible [38], help preserve the user's mental map with interactively changing layouts [30], and impose domain-specific constraints e.g., for biology [4, 12]. Spatially ordered treemap layouts [16, 24, 65] take into account the relative spatial position of each node and the distance between the nodes to create a treemap visualization. Spatial ordering reduces the cognitive load associated with finding a node based on its location and supports identification of relationships and trends [65]. In the case of cerebral arteries, anatomical and spatial context is crucial for many diagnoses. Thus we use a novel constraint-based layout to help preserve this context while conserving screen space. Generic constraint-based algorithms [53] and their implementations [62] can impose established domain-specific constraints on the network layout. In our case, the network constraints were not well established and emerged only through the design iterations (see Sec. 6). Therefore, we did not use generic constraint-based algorithms [53] at the start of our prototype.

Mixed Hierarchy Network Layout: Networks with both hierarchical and non-hierarchical components—e.g., UML diagrams and cerebral artery systems—are called mixed hierarchy networks. Gutwenger et al. [29] developed a technique for visualizing mixed hierarchy UML diagrams. To visualize the Circle of Willis we leverage a key feature they propose: that undirected edges should run horizontally while directed edges run vertically and monotonically.

Take-away: Our work builds on existing network and tree layout techniques and adapts it for the context of an arterial network. We use constraint-based and mixed hierarchy layout techniques to ensure key readability criteria are met. See Sec. 7 for more detail.

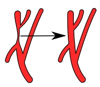
Domain Goals	Stage (I)nterview (O)bservation I	O	Graph Task Abstraction[40]	Analytical Task Abstraction [08] {action, target}	Abstract Task Description
1. Stenosis Detection 	X	X	Browsing (Follow path) Attribute (Edge attribute)	{locate, outlier}	Stenosis detection is a **browsing task**, where the user **follows a path** from the bottom of the network and performs an **attribute based** comparison of the link/edge weight to **locate** an **outlier** artery with significantly lower weight than its connected edges.
2. Aneurysm Detection 	X	X	Browsing (Follow path) Attribute (Edge attribute)	{locate, outlier}	Aneurysm detection is a **browsing task**, where the user **follows a path** from the bottom of the network and performs an **attribute based** comparison of the link/edge weight to **locate** an **outlier** artery with significantly higher weight than its connected edges.
3. Find Missing Arteries 	X	X	Overview	{explore, similarity}	Find missing arteries is an **overview task**, where the user compares two sub-networks of the main network and visually **explores similarity** between the sub-networks.
4. Blood Flow Analysis 	X		Browsing (Follow path) Attribute (Edge attribute)	{locate, feature}	Blood flow analysis is a **browsing task**, where the user **follows a path** from the bottom of the network and performs an **attribute based** comparison of the link/edge color coding to **locate** a **feature** where there is a sharp change in the color coding between the parent-child edges.
5.Therapeutic Planning 	X		Topology(Accessibility) Browsing (Follow path)	{browse, topology}	Therapeutic planning is a **topology task** where the user determines accessibility of stenosis/aneurysm site from the bottom of network by **following path**. Users are aware of the target and are searching for the best way to get to the location, it can be also categorized **browsing** of **topology**.

Fig. 4: Summary of domain goals with accompanying abstract graph and analytical tasks.

4.3 Abstraction with Context

We can simplify a visualization of spatial data by abstracting away unnecessary complexities of the data. For example, Grabeler et al. [26] generated tourist maps of selected areas that highlight key landmarks but hide details about streets and locations which are of no interest. Often, abstractions supplemented with context can assist the user in understanding the representation [65,68]. We can provide the context in spatial data by preserving relative spatial position in the visualization. Subway route maps, e.g., in Boston, USA [44] and Delhi, India [14], were designed to present routes as clearly as possible using abstraction but preserved some spatial context such as the relative positions of routes and landmarks. Another method to provide context in an abstract representation is to associate it with the true spatial information. For example, Yang et. al. [68] developed an abstract NodeTrix [31] representation to visualize neural connectivity in the brain. In their representation, the NodeTrix visualization was overlaid on a schematic representation of the brain to provide context. In another example, Dykes [18] demonstrated the use of linked views to provide geospatial context with an abstract visualization.

Take-away: We use abstraction with context in our design to provide spatial context for the abstract topology visualization of cerebral arteries.

5 Domain Goals and Task Abstraction

In order to better understand the diagnostic tasks (domain goals) and workflow in neuroradiology for cerebrovascular diseases, we worked closely with a neuroradiologist at the Brigham and Women's Hospital (24 years of experience) and a neurological MRI physicist at the Dana-Farber Cancer Institute (10 years of experience). Both experts are co-authors of this paper. In order to generate the domain goals we conducted a series of interviews with these experts, solicited their feedback in our iterative design process, and further validated the curated domain goals through a series of observational studies.

Expert Interviews: We conducted a series of 10 open-ended conversational style interviews with the 2 experts which took on average 90 minutes each. The open-ended style gave our experts the freedom to provide us with sufficient introduction and supporting information from the field of neuroradiology, discuss their current diagnostic practices, and share the pros and cons of the existing techniques.

Expert Observation: We conducted 4 observational studies with the 2 experts at Brigham & Women's Hospital with each observation session lasting on average 6 hours. For the studies we followed a shadowing methodology, a common observational procedure for premed students to understand a physician's typical work activities in a clinic or hospital setting [52]. In requirement analysis literature [25] shadowing resembles a protocol analysis procedure. In protocol analysis, the analyst observes experts in their natural workflow and the experts talk out loud and explain their tasks. This protocol analysis procedure worked well with our experts' workflow as the expert already narrates the diagnostic process aloud, transcribed in realtime with dictation software, for their required radiological case study report. Shadowing importantly enabled us to uncover domain goals that were not shared by the expert during the interview procedure.

5.1 Domain Goals

Based on hand-written notes collected during the expert interviews and observational studies, we were able to apply open-coding and summarization techniques to identify the expert's domain goals for diagnosis of cerebrovascular diseases in medical images. All domain goals are applicable to diagnoses with either CTA or MRA imaging datasets. In ranked order of importance, the observed domain goals are:

1. Stenosis Detection: Identify an intracranial stenosis, i.e., narrowing of an artery inside the brain. Over time, fat can be deposited along the walls of medium and large arteries in the body, causing them to become narrowed or even blocked. To detect a stenosis experts use artery visualization to look for abnormally narrow arteries.

2. Aneurysm Detection: Detect a brain aneurysm, which is an abnormal widening or ballooning of a cerebral vessel. Brain aneurysms occur when an injury, congenital disability, or other diseases weakens the wall of the vessel. To detect aneurysms experts use artery visualizations and look for the balloon-like bulge of artery walls. The doctors additionally examine the 2D source image 'slices' to look for potential bleeding in the brain, which is not detectable in artery visualizations.

3. Find Missing Arteries: The distribution of arteries in the left and the right hemisphere of the brain should be fairly symmetrical for a healthy patient. A highly uneven distribution of arteries in the left and right hemisphere may be due to a blockage (clot) which obstructs blood flow. These structural differences between the left and right sides (vascular malformations) can be indicative of this artery occlusion, a common indicator of stroke [61]. If no blood is flowing in the arteries, then it will not be visible in the images thus doctors look for "empty" or missing branches in the artery visualization.

4. Blood Flow Visualization: Blood flow volume data is very useful for the detection of cerebrovascular diseases since major vascular abnormalities are accompanied by a disruption of blood flow. For the detection of these diseases, a blood flow visualization should enable the differentiation between regions of regular and irregular blood flow. Currently this data, either calculated through a hemodynamic blood

© 2019 IEEE. This is the author's version of the article that has been published in IEEE Transactions on Visualization and Computer Graphics. The final version of this record is available at: 10.1109/TVCG.2019.2934402

flow simulation or interpolated from the luminosity of the contrast dye, is not readily available to radiologists in their clinical imaging suites. Experts sometimes try to qualitatively interpolate this information from the luminosity of voxels in the artery visualization or raw imaging data.

5. Therapeutic Planning: To treat cerebrovascular diseases, interventional radiologists and surgeons repair diseased arteries through invasive methods including stent insertion or balloon angiography in order to widen the artery and restore blood flow. These interventional procedures require careful planning and execution. An essential part of interventional surgical planning is the ability to carefully navigate the surgical equipment to the site of stenosis and aneurysm in the arteries, guided through medical images and artery visualizations of the patient.

5.2 Task Abstraction

As discussed in the preceding section, the primary neuroradiology domain goals involve finding abnormalities in the network of cerebral arteries. Following conventional design study and nested model procedures [46, 56], we frame the domain tasks as low-level analytic tasks both for generalizability as well as for an aid in visual encoding design choices. In this paper we use utilize the Brehmer and Munzner multi-level task typology [8] to determine the *{Action, Target}*. As we established in Sec. 3, the cerebral arteries can be defined as a network. Here we contribute a network task abstraction of the neuroradiology domain goals using the Lee et al. graph task taxonomy [40]. For example: *"Stenosis detection is a browsing task, where the user follows a path from the bottom of the network and performs an attribute based comparison of the link/edge weight."* These network task taxonomy classifications were essential for developing our novel network layout for the CerebroVis. The domain goals, network tasks, and general analytical tasks are summarized in descending order of importance for a diagnosis in Fig. 4. Both the general abstract and network task definitions helped ensure that the final tool and layout supported the domain goals.

6 CerebroVis Design Process

We developed CerebroVis in a user-centered iterative design process with domain experts in order to ensure validity, accuracy, and applicability of our final design. We also relied on our abstract and network task analyses to ensure appropriate choice and visual design of our final 2D encoding. In the following sections we discuss our design iterations (see Fig. 5) and summarize the design requirements in terms of both visual encoding language as well as network topology.

6.1 Iterative Design Process and Goals Formulation

Iteration 1—3D Rendering: In our first design iteration we focused on the development of a clean and clear 3D rendering, close to conventional approaches. We used Sharkviewer [32], an existing open source tool, to create a 3D visualization of the cerebral artery structure (Fig. 5 (1)). Although conventional techniques use both volume rendering and isosurface techniques, in both cases the goal is to produce a rendering with clear hard surfaces and structure. The rendering is constructed using the spatial information provided in the digital segmentation data (Sec. 8). The visualization is interactive and the experts can pan, zoom, and rotate. This rendering was also used in interviews to discuss the existing diagnostic techniques with experts (Sec. 4).

Expert feedback: The experts like the 3D rendering as it preserves the spatial and anatomical properties of the arteries. However, in order to maximize efficiency and minimize cognitive load, the experts prefer views with minimal or no interaction. Most importantly, the experts noted that the 3D rendering suffers from issues of occlusion and is liable to inaccurately render small merging or tangled geometries (i.e., small or tangled features are merged into single 3D feature). The 3D rendering also proved difficult to use with diagnostic tasks that require "path following" to trace a particular artery through the 3D space.

Iteration 2—2D Tree Diagram: In order to counter the issues of occlusion and rendering artifacts, support path following tasks, and eliminate interaction with the visualization, we created an abstract 2D orthogonal tree diagram visualization of the arteries. To make the pseudo-hierarchical artery structure readable, we imposed a binary tree structure on the network. Also, in order to preserve some spatial components of the artery geometry in support of stenosis and aneurysm detection tasks, we encoded artery width and length in the edge width and length.

Expert feedback: Although the 2D tree diagram eliminated the need for interaction, it had several major drawbacks including the lack of apparent symmetry preservation between the left and right hemispheres, overemphasis on (normal/healthy) artery sub-trees with very long branches, and the inability to easily compare the widths of arteries due to the varying lengths. This last point in particular led to the experts inability to identify stenoses, aneurysms, and missing arteries.

Iteration 3—2D Tree Diagram without Artery Length Encoding: In the third design iteration we made two major changes to the visual encoding: a new tree rendering style and a new tree layout. For the tree drawing, we moved from an orthogonal to arced tree (Fig. 5, Iteration 3a), which was laid out using the Reingold-Tilford Algorithm [50]. For consistency, we use the implementation provided by the D3.js library [7]. Also, in order to better support comparison of artery widths for diagnosis of stenosis and aneurysm detection we removed the artery length encoding. In addition, the arced style more closely resembles the real physical appearance of arteries as compared to the sharp right-angles of the orthogonal layout and is easier for path following as discussed in Sec. 4. Finally, we developed an alternative arrangement of the arteries (Fig. 5, Iteration 3b) specifically designed to allow visual comparison of the symmetry between the left and right sides of the brain. In this view each tree depth is at the same vertical level, and in the horizontal arrangement branches are pivoted to encode some aspects of spatial position on the left or right side of the brain.

Expert feedback: The new arced style for the tree diagram was much appreciated for its clarity, easy path following, and ability to compare artery widths. The new layout also more accurately presented the balance between the left and right hemispheres. However, the tree representation did not accurately present the network data near the base of the system, in particular the Circle of Willis (CoW, see Sec. 3). The CoW is a key anatomical feature of the cerebral artery system and also provides an important spatial point of reference for overall layout interpretation.

6.2 CerebroVis Design Requirements

Based on our task abstraction (Fig. 4) and expert feedback from our design iterations (Sec. 6.1), we developed several design requirements to inform our final design:

DR1. *Preservation of expert mental model:* Through our iterative design we established that certain vascular structures, spatial layouts, and visual cues are necessary to provide sufficient context for accurate interpretation. Purchase et al. [49] found that as a user observes and understands the layout of a graph, she creates an internal representation of the information about the data as conveyed in visual forms. For a neuroradiologists, identification of the arteries of the Circle of Willis (CoW), along with its connected arteries' shapes and sizes, is essential. Therefore, to provide sufficient anatomical context we divide this design requirement further into:

a. Distinct and consistent representation of the CoW.
b. Preserve position of cerebral arteries relatively to the CoW.
c. Preserve natural variability within the cerebral arteries.

These design requirements allow users to read and interpret the network visualization.

DR2. *Highlight abnormalities:* Identification of abnormal topology, geometry, and attributes of the cerebral arteries is the primary domain goal as established by our task analysis. These diagnostic tasks heavily informed the iterative design as discussed in Sec. 6.1. The specific design requirements to support the identification of these abnormalities include:

a. Provide a readable network visualization of the cerebral network.
b. Display abnormal narrowing or widening of the arteries.
c. Compare topology between left and right cerebral arteries.
d. Show direction and volume of blood flow within the arteries.

These design requirements support abnormality detection domain goals (Fig. 4 (1–4)).

DR3. *Help experts gain confidence with interpreting an abstract encoding:* Once an expert has identified an abnormality indicative of disease in the cerebral arteries, disease treatment and intervention procedures need to be determined and executed. In this follow-up step an expert needs to identify the abnormality in the 3D rendering as well as original imaging data. In order to make the network layout interpretable with sufficient context this design requirement consists of:

a. Allow examination of abnormal geometry.
b. Locate the abnormality in the 3D rendering.

Ability to locate and examine an abnormality plays a critical role in therapeutic planning procedure (Fig. 4 (5)).

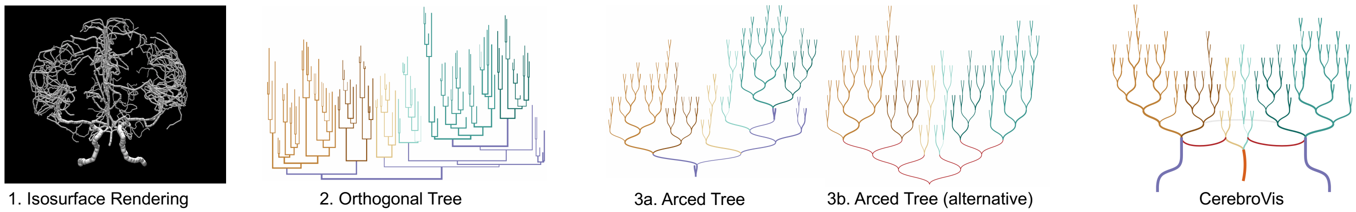


Fig. 5: The visual design evolution of CerebroVis from (left) 3D rendering to (right) final 2D network representation.

7 CEREBROVIS

7.1 Spatially Contextualized Network Layout

As illustrated in Fig. 1, the cerebral network consists of two subnetworks: the Circle of Willis (CoW) (an undirected weighted cycle) and the directed binary trees of the cerebral arteries. In the following sections we discuss the layout and design of each subnetwork separately.

Circle of Willis (CoW): The CoW is an undirected weighted cyclic network composed of the P. Comm., A. Comm., Internal Carotids (IC), Basilar Artery (BA) and parts of Anterior Cerebral Artery (ACA) (Sec. 3). To represent the accurate cyclic structure of the CoW we arc the A. Comm. upward and the P. Comm. arteries downward as shown in Fig. 6 (Accurate Network Topology). Any missing arteries in the CoW are represented using a dashed line. In our data [67] the A. Comm. was excluded to maintain a binary tree structure thus a dashed line is used to represent the artery (see Fig. 6).

To match the internal representation of the cerebral network and preserve the user's mental model, we abstract the geometry of the IC and BA (see Fig. 6: Abstract Geometry). Abstraction of the geometry of the carotid arteries is a two step process: we first find the length and the total number of bends. To estimate the bends we use the spatial position of each segment that makes up the IC. In Fig. 6 (Abstract Geometry), there are a total of 3 bends. The bends are broken down as S1–S3 in the Fig. 6 for explanation. In S1 and S3 the artery moves in a vertical direction, therefore a change in the y vertical direction is more than change in the x direction. Next, a Beizer curve is generated with an equal number of bends and the length of the curve is proportional to length of S1–S3 in the carotid arteries. Additionally, the width of each artery curve is proportional to the width of the artery width in the original data. **Requirement Satisfied: *DR1a***—We distinctly and consistently represent the cyclic network structure of CoW and the shape of the carotid and basilar arteries.

Global Spatial Position of Cerebral Arteries: Global spatial preservation operates at two levels. The first level divides the left and right hemispheres of the brain. This level ensures that cerebral artery trees on the left and right never cross over, which would be anatomically inaccurate. Within each hemisphere the arteries are located in specific positions. For example, from the center moving outward we place the PCA, ACA, then MCA (a schematic of this is presented in Fig. 1: Spatial Constraints). This positioning scheme also ensures that cerebral trees maintain familiar position with respect to the CoW. **Requirement Satisfied: *DR1b***—Spatial constraints restrict the placement of the cerebral artery trees to a deterministic area of the display and preserve position relative to the CoW. Our validation (Sec. 4) confirms that this design enables experts to distinguish between cerebral trees.

Local Spatial Position of Cerebral Arteries: Local spatial preservation is tied to the aspect of safeguarding spatial context within each cerebral tree (PCA, MCA, and ACA). Each artery in the original data exists as a 3D geometry. In CerebroVis, we preserve approximate spatial (x,y) positions for each artery of the cerebral trees. Vertical height y of each artery is represented by the height/depth of the edge in the cerebral tree. The x position of arteries is preserved relative to their branching site. For example, in Fig. 7 (Original Order) the original artery order from the source data is presented. If the original order does not match the the relative spatial position of any two arteries, as is the case in Fig. 7 (Spatial Position) where artery c is to the left of artery a, the ordering is swapped (Fig. 7 (Updated Order)). This process is repeated from leaf nodes to the root which ensures that the arteries closer to the CoW are positioned closer to the CoW and otherwise away from the CoW on the x axis. **Requirement Satisfied: *DR1c***—Local position

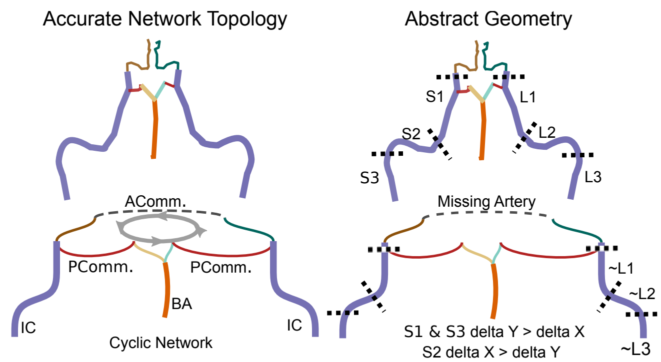


Fig. 6: Left: Reconstructing of the Circle of Willis (CoW) cycle of the cerebral artery network. Right: How CerebroVis abstracts the carotid artery geometry to preserve a frame of reference.

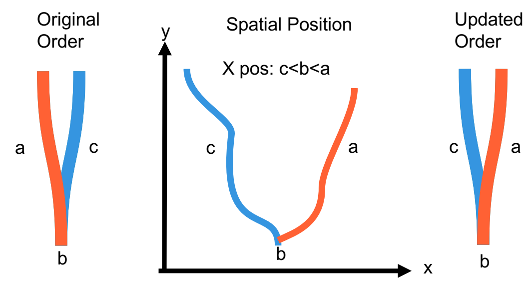


Fig. 7: At each artery bifurcation CerebroVis preserves the relative spatial context of each subtree by comparing average horizontal position.

preservation ensures that when experts look at an edge in the network they have an approximate sense of the position of the artery in the brain.

Layout Technique: Our novel layout technique CerebroVis, outlined in Fig. 8, uses the data discussed in the data section (Sec. 8) to create the representation. CerebroVis uses a layered upward planar drawing by DiBattista et al. [5] which enables a fast lookup of the hierarchy [9], eases topology comparison, and supports the experts' mental model of viewing the brain from the perspective of orthogonal faces of the data. The implementation of the technique is available at https://osf.io/e5sxt/

7.2 Visual Encoding to Highlight Abnormalities

Based on the domain goal analysis (Sec. 5) and iterative design goal formulation (Sec. 6.1), CerebroVis needs to provide an easily readable network visualization with a visual encoding suitable for the following tasks: (1) locate outlier artery shape (stenosis and aneurysm), (2) explore similarity to find missing arteries, and (3) locate a group of arteries with abnormally low blood flow.

Readable Network Visualization: In support of our design goals, CerebroVis presents a readable network layout inspired by literature in network visualization (Sec. 4). The final layout (Fig. 1) minimizes edge overlap, replaces long curved edges with small arced edges, and distinguishes between direction (uni-directional and bi-directional) network edges as explained in the network layout section (Sec. 7.1). The reduced diversion due to branch disentanglement represents accurate

© 2019 IEEE. This is the author's version of the article that has been published in IEEE Transactions on Visualization and Computer Graphics. The final version of this record is available at: 10.1109/TVCG.2019.2934402

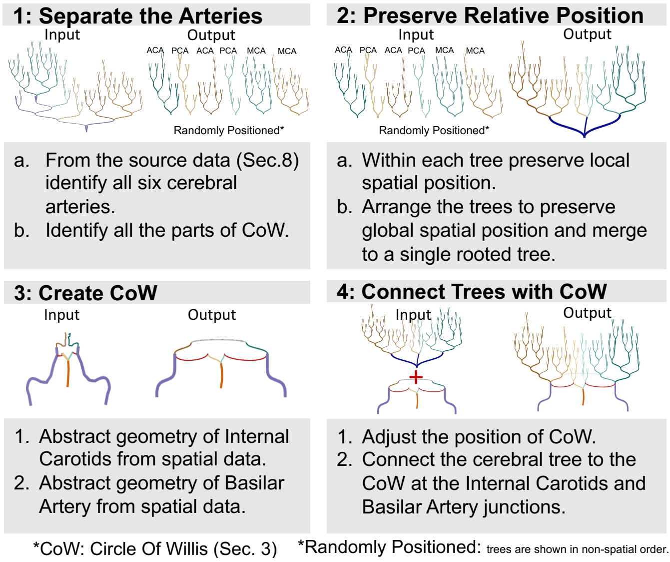


Fig. 8: CerebroVis converts an input of a single rooted tree into a mixed hierarchy network visualization via four steps.

network topology and reduces the cognitive load for path following in the cerebral network. Minimized edge overlaps enable the occlusion-free inspection of each artery. **Requirement Satisfied: *DR2a***—We provide a readable visualization of the cerebral network.

Locate Outlier Artery Width: CerebroVis visually encodes the width of the artery. Network edges are scaled proportional to the average width of the representative artery. Arteries linearly taper (narrow) as they move away from the Circle Of Willis. Thus we use a linear scale to map the width of the arteries to network edge weights. Whenever an artery significantly deviates from the natural linear tapering and appears abnormally narrow or wider than its parent and child arteries (Fig. 10), those arteries can be potential sites of stenosis and aneurysm in the cerebral network. **Requirement Satisfied: *DR2b***—We display abnormal narrowing or widening of the arteries.

Explore Similarity: Topology comparison of the cerebral arteries is an overview task for the neuroradiologists (Sec. 5). To compare topologies, experts estimate depth and width difference between the left and right arteries (PCA, ACA, and MCA). CerebroVis enables depth comparison between left and right arteries by placement of branching sites at the same vertical height. For example, in Fig. 9 (1) the MCA on the right and the left begin at the same vertical height. Additionally, each subsequent branching site is positioned at the same vertical height. Therefore, the difference between the left and right trees can be determined through comparison of the vertical positions of leaf arteries. The comparison of width is possible at an overview level as the arteries take space proportional to their tree width. For subtle differences, at each bifurcation site the length of an artery roughly encodes the width of the subtree at that bifurcation. For instance, in Fig. 9 (1), the MCA on the left has a greater width than the MCA on the right. **Requirement Satisfied: *DR2c***—We enable comparing topology between left and right cerebral arteries.

Analyze Blood Flow: In CerebroVis an edge also encodes the amount of blood flow in an artery. To encode the flow, we use a linear color scale between and (see Fig. 10). To test the visual output of the encoding, we simulate blood flow in the cerebral arteries with a simple linear model. The model divides a fixed amount of blood flow through the arteries, where the flow in an artery is proportional to its width and inversely proportional to the height of the artery from CoW. The model is discussed in the Supplemental Material. The output from the blood flow volume model with a simulated thrombosis or blockage in an artery, shown in Fig. 10 (Abnormal Flow), demonstrates how the same set of arteries will show a sudden white outlier indicating a blockage. We believe complex blood flow models can also be represented with CerebroVis, and we envision testing more models in future. **Requirement Satisfied: *DR2d***—We show the direction and volume of blood flow within the arteries.

7.3 Linked Views to Maintain User Confidence

CerebroVis preserves context with its spatially constrained layout. However, it does not preserve the true spatial position and 3D geometry of an artery. To provide anatomical context, we created an accompanying interactive dashboard which uses a linked view approach between edges of CerebroVis and a 2D projection of a 3D visualization (Fig. 9). The projection can be changed to show arteries in the conventional radiology views from the front, top, and side of the head. For linking, a user is able to click on any edge in the network and the corresponding artery in the 3D projection is highlighted (Fig. 9 (2)). The highlighted 2D projection artery can be used to validate the geometry and location of an artery in the brain.

To identify and distinguish cerebral arteries easily in the 2D projection and CerebroVis network visualization, we use a categorical color map. This color coding uses the same hue for arteries on the same side of the brain, but varies the saturation of the color. The saturation captures the depth perspective of the arteries. For example, ACA provides blood to the front of the brain so they are darkest and PCA provides blood to the back of the brain and they are lightest. In the CoW each type of artery receives a unique color, with a bright red hue for P. Comm., to highlight it in the 2D projection.

7.4 CerebroVis Web Application

An open source web-based implementation of CerebroVis (Fig. 9) is available at aditeyapandey.github.io/CerebroVisProject We use D3.js [7] for the implementation of CerebroVis. Additional details about the tool are available in the Supplemental Material.

8 Data and Implementation Robustness

For the design, development, and evaluation of CerebroVis and the new 2D layout we used a collection of open source MRA datasets of 61 healthy patients from Wright et al. [67]. We chose this collection of data as it is representative of real clinical data (i.e., ecological validity) both in terms of imaging modality (MRA) and quality. The data is also freely available and has had all identifiable patient information properly removed. Each dataset additionally includes the 3D cerebral artery geometry extracted from the MRA scan using a conventional medical imaging segmentation technique [15]. The segmentation extracts a binary tree stemming from the Basilar Artery (BA). Starting from the BA, all visible connecting vessels are added with the exception of the anterior communicating arteries in order to maintain a binary tree structure.

Our data ingestion process is discussed in detail in the Supplemental Material but we provide an overview here. The vascular structure data is provided as an swc tabular file—a format originally developed for neural connectivity networks [32]. In the file each artery is represented by a chain of artery segments, each of which is described by its position, size, and parent/child edge relations. We convert the swc data into a rooted tree to show the hierarchical structure of the cerebral arteries which we can then modify to create a mixed network that includes the Circle of Willis cycle.

To validate our data conversion pipeline, network layout technique, and visualization design we loaded and visualized the 61 scans from Wright et al. [67]. We manually assessed the robustness of the conversion according to three criteria: (1) *Visual Similarity of the CoW (Circle of Willis)*: In each scan we verified the similarity of the shape of CoW in the 3D projection and CerebroVis. (2) *Spatial Constraints are Imposed Properly*: In each scan we examined the position of the cerebral artery branches relative to the CoW then randomly chose one branch to traverse and validate artery positions using the linked views. (3) *Views are Linked Accurately*: We randomly selected 20 scans, for which we interactively traversed the entire cerebral artery system using the CerebroVis Linked Views. In all cases artery marks in the 3D projection were linked properly to their counterparts in the 2D projection. **Rationale**: Criteria 1 and 2 validate our primary design goals (discussed in Sec. 6.2). Criteria 3 validates the accuracy of data binding between the 3D projection and CerebroVis visualization.

9 Comparative Evaluation

To validate and evaluate the design and functionality of CerebroVis against a conventional 3D visualization we conducted a mixed-methods study [34]. The evaluation included a controlled task-based experiment where experts diagnosed a simulated intracranial artery stenosis (narrowing of the artery) and a semi-structured interview.

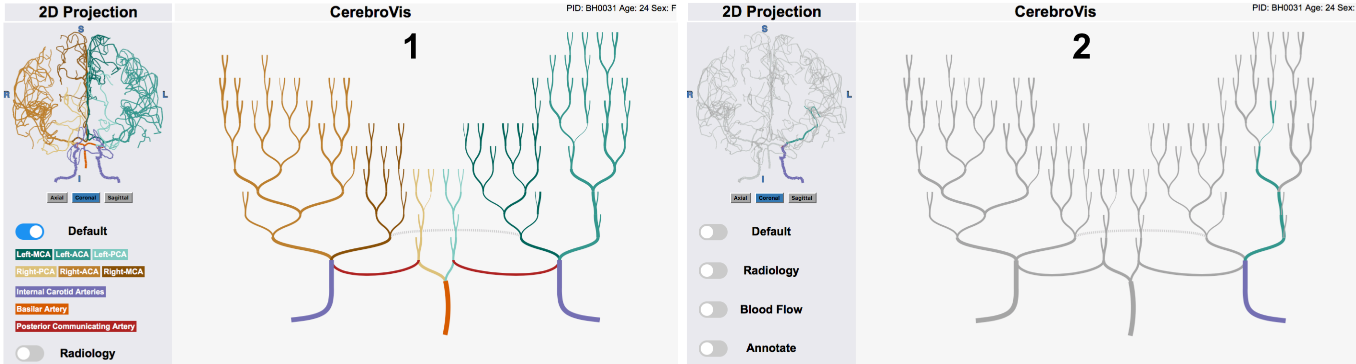


Fig. 9: The CerebroVis Dashboard with categorical coloring to differentiate arteries. Left: A cerebral artery scan with a stenosis in the MCA ■. Right: Users can click on an artery mark in CerebroVis and the corresponding mark is highlighted in the 2D projection. This feature allows users to validate the stenosis with the underlying geometry and plan for therapeutic surgery.

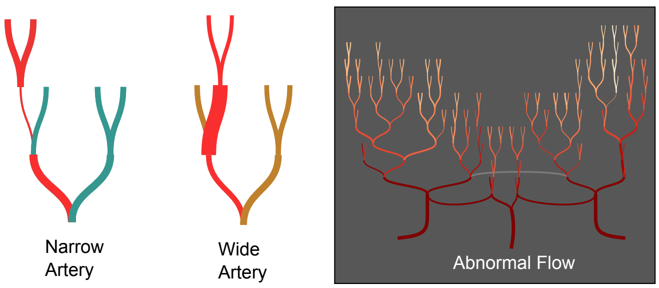


Fig. 10: Left: Examples of an abnormally narrow artery ("stenosis") and wide artery ("aneurysm"), with the relevant branches colored red ■. Right: Example of a blood flow color encoding with a blockage disrupting normal flow of blood with blood flow colored on a scale between □ and ■.

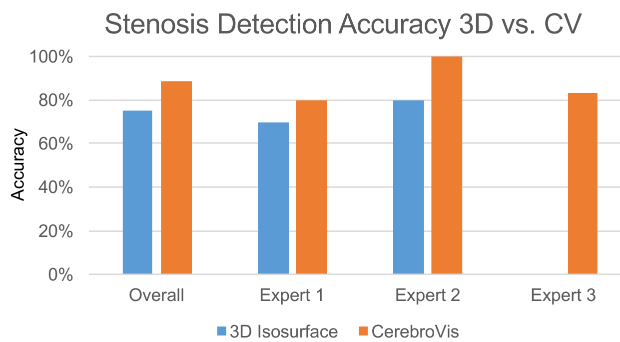


Fig. 11: Expert accuracy at identifying simulated stenoses using the 3D visualization (3D) and 2D network layout CerebroVis (CV). One expert (#3) did not complete the 3D visualization condition due to early departure from the interview.

9.1 Participants

For our study we recruited three neuroradiologists from Brigham & Women's hospital who had no prior experience with CerebroVis. They had 8, 25, and 40 years of experience. In addition to their participation in our evaluation, they provided additional post-experiment feedback on this research and thus are included as co-authors on this paper.

9.2 Methodology

We arranged an evaluation session with each participant. Each session lasted at minimum an hour, with some of longer duration based on expert availability for an extended interview to solicit additional qualitative feedback. Participants spent ~15 mins on the introduction and tutorial, but as the tutorial was interactive it often lasted longer than planned. The next ~25–30 mins were spent on a within-subjects task-based controlled experiment which simulated intracranial artery stenosis diagnosis tasks. We chose stenosis diagnosis because it is a primary cause of cerebrovascular disorders [10]. Finally we performed ~15 mins of semi-structured interview to solicit qualitative feedback on features, usability scenarios, drawbacks and disadvantages, and potential directions for future work.

9.3 Stimuli and Tasks

We used as our stimuli two black-and-white cerebral artery visualizations: a 3D visualization (3D) (Fig. 2 (C)) and the CerebroVis (CV) 2D network (Fig. 2 (D)). We chose a black-and-white color encoding to emulate the conventional clinical visualizations. The 3D isosurface visualization similarly echoed the clinical convention of limited interaction with only clockwise/counterclockwise rotation of variable speed around the vertical axis enabled (i.e., no panning, zooming, or free-form rotation). To control the rotation, we provide options including stop the rotation, change the default direction of rotation, and alter the speed of rotation. In the 2D network representation, users were only shown the static novel 2D CerebroVis layout without the linked view or context. As our data collection was of presumably healthy patients (Sec. 8), we simulated stenoses in 5 random data sets under the close guidance of domain collaborators. All of the injected stenoses were severe with over 70% narrowing of the artery. We induced only severe cases because they pose a greater risk to life and are the top priority for clinical diagnosis. The location of the stenoses were randomly determined but were restricted to the cerebral branches PCA, MCA, and ACA. The altered scans were then randomly mixed with 5 unaltered scans. Each participant was asked to read the 10 scans in a randomized order with one visualization, then switched to the other visualization to read the same scans again. They were asked to identify any stenosis they detected. One participant was not able to finish the controlled experiment as the tutorial lasted longer than expected and they were paged to attend to a critical clinical case. We have 6 answers for CV and none for 3D for this participant.

9.4 Quantitative Analysis and Results

We analyzed the $n = 46$ (CV= 26, 3D= 20) answers from our three neuroradiologists using simple risk statistics, as we did not have enough participants to conduct a more rigorous statistical analysis. We compare the stenosis detection accuracy for the control 3D visualization (3D) and treatment CerebroVis (CV) in Fig. 11, overall and for individual experts. Below we detail the results of our quantitative analysis including the control absolute risk (ARC), treatment absolute risk (ART), and the absolute risk difference (ARD). Note that due to the small sample size these analyses should be interpreted as formative results. Our data and analysis code is available at osf.io/e5sxt

In general, we found that the neuroradiologists were more likely to correctly answer the diagnostic task questions using the 2D CerebroVis layout compared to the 3D visualization *(ARC 25%, ART 12%, ARD 13%)* and more likely to correctly identify an introduced stenosis, i.e. a true positive (*ARC 40%, ART 7%, ARD 33%*). However, participants

© 2019 IEEE. This is the author's version of the article that has been published in IEEE Transactions on Visualization and Computer Graphics. The final version of this record is available at: 10.1109/TVCG.2019.2934402

were also more likely to identify false positives with CerebroVis, i.e., a feature the participant believes is a stenosis but that we did not introduce *(ARC 10%, ART 17%, ARD -7%)*. We reflect on these results in the discussion (Sec. 10.2).

9.5 Qualitative Results

We observed two consistent recurring themes in our interviews:

The linked view feature of the CerebroVis dashboard is essential to validate the identified abnormality with source data: Participants were excited about the interaction between CerebroVis and the 2D isosurface projection of the brain arteries. One of the participants said, *"If I want I can spread out the arteries in 2D and then conveniently go back to the real data."* Another participant identified that the 2D isosurface projection was insufficient alone, but since the tool can link the abstract artery mark to the exact spatial position they could verify geometry in the source data. They also said, *"None of the existing visualization technique has the feature to link to the source data."*

The 2D layout has the potential to be an alternative to 3D representation: All three participants saw potential in this novel cerebral artery visualization. According to one, *"This is already done by cardiologists to visualize stenosis, I am surprised we don't do it."* Another mentioned, *"This is like a panoramic view of the arteries, which is an interesting engineering design."* When discussing cost-effectiveness one said, *"2D visualization is probably going to be faster than the 3D visualization."*

10 Discussion

10.1 Abstraction with Context

Through our design process we realized that the visualization of complex systems, like the cerebral artery network, can be intrinsically complicated due to the complex nature of the underlying data. In our iterative design process, we learned that the visualization of complex systems can be abstracted to represent information pertinent to domain goals and abstract tasks. For example, in a stenosis detection task, the information about artery width and the network topology outweighs the importance of features like artery length and exact spatial position of the arteries. To reduce the visual complexity, we designed a network representation which displayed information about the artery width and the topology of the cerebral branches (Fig. 5 (3)) and de-emphasized non-required features such as artery length and the exact spatial position of arteries. While the experts appreciated the design, they did not understand how to interpret the non-spatially contextualized tree visualization back into 3D space. Ultimately, in this design study, we learned that abstract visualizations of complex systems could improve the task performance but there is a need to supplement the abstraction with user-centered context. During our design evolution, we found two visualization design paradigms that helped us embed context in CerebroVis:

1. Spatially Constrained Network Layout: The cerebral artery network representation is spatially constrained to preserve the relative position of the arteries as per their location in the brain. The constraint provides a spatial context for the abstract topology visualization of the cerebral arteries. In our evaluation, we observed that the spatial context of cerebral arteries assisted the neuroradiologists in interpreting the abstract visualization. For example, in the stenosis detection task, experts correctly identified the site of the stenosis without the use of any legend. Thus user-centered spatial constraints embed the necessary context to understand the abstract representation of a complex artery network.

2. Linked Views: Linked views can be used to explore multiple facets of a dataset [51]. In CerebroVis, the abstract visualization of the cerebral arteries does not preserve the exact spatial position and 3D geometry. Therefore, to preserve the 3D anatomical context of the data, the spatially constrained network visualization is linked with the 2D spatial projection of the brain data in our tool implementation. The link between the 3D spatial brain and abstract network visualization is essential for the clinical domain experts as it enables them to examine the geometry of the deformed artery in full context. For example, we do not encode the bends and curves of the arteries, but this information plays an important role in the treatment of vascular abnormalities. The availability of anatomical context through linked views allows an expert to validate the abnormality and thereby instills confidence in using the novel visual representation.

Design Recommendations: In this design study we introduced a novel method to visualize the human cerebral artery network. While our visualization is tied explicitly to the human cerebral system, the design paradigms discussed in this study can be broadly applied outside of the brain to visualize other hierarchical circulatory systems. In this paper, we recommend the use of abstract topology visualization of the circulatory system to support network tasks like path following and symmetry comparison. The familiarity of the abstract topology can be increased with relative spatial constraints to match the internal representation of the user mental model, and the abstract representation can be further linked with the spatially and anatomically accurate visual representation to allow in-depth analysis of vascular abnormalities.

10.2 Evaluation Results

Overall, we found that CerebroVis was easy to understand for all three study participants and they quickly adapted to the design. The formative quantitative evaluation results align with the qualitative feedback: overall CerebroVis was more effective for identifying stenoses in cerebral arteries. Experts did a better job of identifying true positives (simulated stenosis cases) with the 2D network layout of CerebroVis as compared to a 3D isosurface visualization. This validates our visualization against the primary design requirement that the visual encoding should support identification of abnormalities (Sec. 6.2). In the evaluation we also notice that CerebroVis performs worse than the 3D representation for the detection of true negatives. We speculate this is could be for two reasons. First, the novel unfamiliar representation of CerebroVis, and the easy length comparisons it provides, may require additional training for experts to distinguish significant and non-significant differences in artery width. Second, the presumably healthy patient scans we use may include undiagnosed stenoses that we did not introduce.

11 Limitations and Future Work

With CerebroVis we support symmetry comparison through the topology comparison of the cerebral arteries. The task of symmetry comparison in cerebral arteries can be further developed by layout design improvements or analytical integration to include more spatial information. For example, the layout could preserve the exact spatial distance between two arteries, branching sites could preserve the angle of branching, or the network edges could encode tortuosity of the arteries. Spatial information will benefit the overall visual comparison, but the effect on the other tasks such as stenosis detection also needs to be carefully considered. Integrating CerebroVis with automated cerebrovascular abnormality detection systems may be beneficial for approximate tasks including symmetry comparison. Doctors use a rough visual comparison to detect differences, thus an abnormality may elude detection. This could be avoided by providing annotated areas of symmetry variation to serve as cues for doctors to examine the highlighted area. Finally, in support of sometimes small artery-width abnormalities, such as aneurysms, instead of encoding edge thickness proportional to the average width of the entire artery the visual encoding could display a variable-width encoding where needed to highlight such outliers.

12 Conclusion

We present CerebroVis, a novel abstract but spatially contextualized network layout for visualizing cerebral artery networks. We also contribute a novel framing and network theory definition of the cerebral artery system. Through expert interview and observations we characterize the domain goals in an ordered list of importance and present them as abstract visualization and network analysis tasks. We evaluate the layout and the co-developed visualization prototype through a mixed methods study with three neuroradiologists. In a controlled task-based study we found that our abstract visualization improved task performance over a conventional 3D visualization for identifying intracranial artery stenosis. From semi-structured interviews we determined that the inclusion of spatial context helped preserve the users' mental maps of the underlying geometry. More broadly, we believe visualization design for other hierarchical circulatory system components outside the brain could benefit from our design methodology.

Acknowledgments

We thank David Saffo, Laura South, Michail Schwab, Sara Di Bartolomeo, and Yixuan "Janice" Zhang for their valuable feedback. A special thanks to Ian Helgi Magnusson for editing and proof reading help. This work was supported by NSF CISE CRII award no. 1657466.

References

[1] B. Alper, B. Bach, N. Henry Riche, T. Isenberg, and J.-D. Fekete. Weighted graph comparison techniques for brain connectivity analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pp. 483–492. ACM, New York, NY, USA, 2013. doi: 10.1145/2470654.2470724

[2] R. Aviv, J. Mandelcorn, S. Chakraborty, D. Gladstone, S. Malham, G. Tomlinson, A. Fox, and S. Symons. Alberta stroke program early ct scoring of ct perfusion in early stroke visualization and assessment. *American Journal of Neuroradiology*, 28(10):1975–1980, nov 2007. doi: 10.3174/ajnr.A0689

[3] B. Aydın, G. Pataki, H. Wang, A. Ladha, E. Bullitt, and J. Marron. Visualizing the structure of large trees. *Electronic Journal of Statistics*, 5(0):405–420, 2011. doi: 10.1214/11-ejs612

[4] A. Barsky, T. Munzner, J. Gardy, and R. Kincaid. Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE transactions on visualization and computer graphics*, 14(6):1253–1260, 2008.

[5] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall PTR, 1998.

[6] M. Borkin, K. Gajos, A. Peters, D. Mitsouras, S. Melchionna, F. Rybicki, C. Feldman, and H. Pfister. Evaluation of artery visualizations for heart disease diagnosis. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2479–2488, Dec 2011. doi: 10.1109/TVCG.2011.192

[7] M. Bostock, V. Ogievetsky, and J. Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, Dec. 2011. doi: 10.1109/TVCG.2011.185

[8] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Trans. Visualization and Computer Graphics (TVCG) (Proc. InfoVis)*, 19(12):2376–2385, 2013.

[9] M. Burch, N. Konevtsova, J. Heinrich, M. Hoeferlin, and D. Weiskopf. Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2440–2448, 2011.

[10] Centers for Disease Control and Prevention. Cdc stroke information. `https://www.cdc.gov/stroke/types_of_stroke.html`.

[11] M. K. Coleman and D. S. Parker. Aesthetics-based graph layout for human consumption. *Softw. Pract. Exper.*, 26(12):1415–1438, Dec. 1996. doi: 10.1002/(SICI)1097-024X(199612)26:12<1415::AID-SPE69>3.3.CO;2-G

[12] P. Csermely, T. Korcsmáros, H. J. Kiss, G. London, and R. Nussinov. Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. *Pharmacology & therapeutics*, 138(3):333–408, 2013.

[13] R. Davidson and D. Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics (TOG)*, 15(4):301–331, 1996.

[14] Delhi Metro Rail Corporation Ltd. `http://www.delhimetrorail.com/Zoom_Map.aspx`.

[15] I. Despotović, B. Goossens, and W. Philips. Mri segmentation of the human brain: challenges, methods, and applications. *Computational and mathematical methods in medicine*, 2015, 2015.

[16] F. S. L. G. Duarte, F. Sikansi, F. M. Fatore, S. G. Fadel, and F. V. Paulovich. Nmap: A novel neighborhood preservation space-filling algorithm. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2063–2071, Dec 2014. doi: 10.1109/TVCG.2014.2346276

[17] C. Dunne, S. I. Ross, B. Shneiderman, and M. Martino. Readability metric feedback for aiding node-link visualization designers. *IBM Journal of Research and Development*, 59(2/3):14:1–14:16, Mar. 2015. doi: 10.1147/JRD.2015.2411412

[18] J. Dykes. Cartographic visualization. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(3):485–497, 1998.

[19] P. Eades and L. Xuemin. How to draw a directed graph. In *Visual Languages, 1989., IEEE Workshop on*, pp. 13–17. IEEE, 1989.

[20] D. Ferrari and L. Mezzalira. *On drawing a graph with the minimum number of crossings*. Politecnico, 1969.

[21] E. K. Fishman, D. R. Ney, D. G. Heath, F. M. Corl, K. M. Horton, and P. T. Johnson. Volume rendering versus maximum intensity projection in ct angiography: what works best, when, and why. *Radiographics*, 26(3):905–922, 2006.

[22] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

[23] G. Gacs, A. Fox, H. Barnett, and F. Vinuela. Ct visualization of intracranial arterial thromboembolism. *Stroke*, 14(5):756–762, 1983.

[24] M. Ghoniem, M. Cornil, B. Broeksema, M. Stefas, and B. Otjacques. Weighted maps: treemap visualization of geolocated quantitative data. In *Visualization and Data Analysis 2015*, vol. 9397, p. 93970G. International Society for Optics and Photonics, 2015.

[25] J. A. Goguen and C. Linde. Techniques for requirements elicitation. In *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering*, pp. 152–164. IEEE, 1993.

[26] F. Grabler, M. Agrawala, R. W. Sumner, and M. Pauly. *Automatic generation of tourist maps*, vol. 27. ACM, 2008.

[27] J. Guerra-Gómez, M. L. Pack, C. Plaisant, and B. Shneiderman. Visualizing change over time using dynamic hierarchies: Treeversity2 and the stemview. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2566–2575, 2013.

[28] J. A. Guerra-Gómez, A. Buck-Coleman, M. L. Pack, C. Plaisant, and B. Shneiderman. Treeversity: interactive visualizations for comparing hierarchical data sets. *Transportation Research Record*, 2392(1):48–58, 2013.

[29] C. Gutwenger, M. Jünger, K. Klein, J. Kupke, S. Leipert, and P. Mutzel. A new approach for visualizing uml class diagrams. In *Proceedings of the 2003 ACM symposium on Software visualization*, pp. 179–188. ACM, 2003.

[30] W. He and K. Marriott. Constrained graph layout. *Constraints*, 3(4):289–314, 1998.

[31] N. Henry, J.-D. Fekete, and M. J. McGuffin. Nodetrix: a hybrid visualization of social networks. *IEEE transactions on visualization and computer graphics*, 13(6):1302–1309, 2007.

[32] HHMIs Janelia Research Campus. SharkViewer. `https://www.janelia.org/sharkviewer`.

[33] D. Holten and J. J. Van Wijk. Visual comparison of hierarchically organized data. In *Computer Graphics Forum*, vol. 27, pp. 759–766. Wiley Online Library, 2008.

[34] R. B. Johnson and A. J. Onwuegbuzie. Mixed methods research: A research paradigm whose time has come. *Educational researcher*, 33(7):14–26, 2004.

[35] W. Johnson, O. Onuma, M. Owolabi, and S. Sachdev. Stroke: a global response is needed. *Bulletin of the World Health Organization*, 94(9):634, 2016.

[36] A. K. Al-Awami, J. Beyer, H. Strobelt, N. Kasthuri, J. Lichtman, H. Pfister, and M. Hadwiger. Neurolines: A subway map metaphor for visualizing nanoscale neuronal connectivity. 20:2369–2378, 12 2014.

[37] S. G. Kobourov, T. Mchedlidze, and L. Vonessen. Gestalt principles in graph drawing. In *Revised Selected Papers of the 23rd International Symposium on Graph Drawing and Network Visualization - Volume 9411*, GD 2015, pp. 558–560. Springer-Verlag New York, Inc., New York, NY, USA, 2015. doi: 10.1007/978-3-319-27261-0_50

[38] M. Krzywinski, I. Birol, S. J. Jones, and M. A. Marra. Hive plots rational approach to visualizing networks. *Briefings in bioinformatics*, 13(5):627–644, 2011.

[39] C. Kubisch, S. Glaßer, M. Neugebauer, and B. Preim. Vessel visualization with volume rendering. In *Visualization in Medicine and Life Sciences II*, pp. 109–132. Springer, 2012.

[40] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task taxonomy for graph visualization. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pp. 1–5. ACM, 2006.

[41] M. H. Lev, W. J. Koroshetz, L. H. Schwamm, and R. G. Gonzalez. Ct or mri for imaging patients with acute stroke: Visualization of "tissue at risk"? *Stroke*, 33(12):2736–2737, 2002. doi: 10.1161/01.STR.0000041999.64363.B2

[42] A. Lin, R. Agid, S. Rawal, and D. M. Mandell. Cerebrovascular Imaging: Which Test is Best? *Neurosurgery*, 83(1):5–18, 06 2017. doi: 10.1093/neuros/nyx325

[43] R. J. Lipton, S. C. North, and J. S. Sandberg. A method for drawing graphs. In *Proceedings of the first annual symposium on Computational geometry*, pp. 153–160. ACM, 1985.

[44] Massachussets Bay Transport Authority. `https://www.mbta.com/schedules/subway`.

[45] H. Mohammed, A. K. Al-Awami, J. Beyer, C. Cali, P. Magistretti, H. Pfister, and M. Hadwiger. Abstractocyte: A visual tool for exploring nanoscale astroglial cells. 2017.

[46] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, Nov 2009. doi: 10.1109/TVCG.2009.111

[47] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. Treejuxtaposer: scalable tree comparison using focus+ context with guaranteed visibility. In *ACM Transactions on Graphics (TOG)*, vol. 22, pp. 453–462. ACM, 2003.

[48] H. Purchase. Which aesthetic has the greatest effect on human understanding? In *International Symposium on Graph Drawing*, pp. 248–261. Springer, 1997.

[49] H. C. Purchase, E. Hoggan, and C. Görg. How important is the "mental map"?–an empirical investigation of a dynamic graph layout algorithm. In *International Symposium on Graph Drawing*, pp. 184–195. Springer, 2006.

[50] E. M. Reingold and J. S. Tilford. Tidier drawings of trees. *IEEE*

© 2019 IEEE. This is the author's version of the article that has been published in IEEE Transactions on Visualization and Computer Graphics. The final version of this record is available at: 10.1109/TVCG.2019.2934402

Transactions on software Engineering, (2):223–228, 1981.

[51] J. C. Roberts. Exploratory visualization with multiple linked views. In *Exploring geovisualization*, pp. 159–180. Elsevier, 2005.

[52] C. Rothwell, B. Burford, J. Morrison, G. Morrow, M. Allen, C. Davies, B. Baldauf, J. Spencer, N. Johnson, E. Peile, et al. Junior doctors prescribing: enhancing their learning in practice. *British journal of clinical pharmacology*, 73(2):194–202, 2012.

[53] F. Schreiber, T. Dwyer, K. Marriott, and M. Wybrow. A generic algorithm for layout of biological networks. *BMC bioinformatics*, 10(1):375, 2009.

[54] H. Schulz, S. Hadlak, and H. Schumann. The design space of implicit hierarchy visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):393–411, April 2011. doi: 10.1109/TVCG.2010.79

[55] H.-J. Schulz. Treevis. net: A tree visualization reference. *IEEE Computer Graphics and Applications*, 31(6):11–15, 2011.

[56] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE transactions on visualization and computer graphics*, 18(12):2431–2440, 2012.

[57] B. Shneiderman and C. Plaisant. Treemaps for space-constrained visualization of hierarchies. 1998.

[58] S. Skwerer, E. Bullitt, S. Huckemann, E. Miller, I. Oguz, M. Owen, V. Patrangenaru, S. Provan, and J. Marron. Tree-oriented analysis of brain artery structure. *Journal of mathematical imaging and vision*, 50(1-2):126–143, 2014.

[59] K. Sugiyama. *Graph drawing and applications for software and knowledge engineers*, vol. 11 of *Series on Software Engineering and Knowledge Engineering*. World Scientific Publishing Company, June 2002.

[60] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, Feb 1981. doi: 10.1109/TSMC.1981.4308636

[61] M. Tanaka, E. Shimosegawa, K. Kajimoto, Y. Kimura, H. Kato, N. Oku, M. Hori, K. Kitagawa, and J. Hatazawa. Chronic middle cerebral artery occlusion: a hemodynamic and metabolic study with positron-emission tomography. *American Journal of Neuroradiology*, 29(10):1841–1846, 2008.

[62] Tim Dwyer. cola.js:constraint-based layout in the browser. `https://ialab.it.monash.edu/webcola/`.

[63] J. W. Wallis, T. R. Miller, C. A. Lerner, and E. C. Kleerup. Three-dimensional display in nuclear medicine. *IEEE Transactions on Medical Imaging*, 8(4):297–230, Dec 1989. doi: 10.1109/42.41482

[64] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive measurements of graph aesthetics. *Information visualization*, 1(2):103–110, 2002. doi: 10.1057%2Fpalgrave.ivs.9500013

[65] J. Wood and J. Dykes. Spatially ordered treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1348–1355, Nov 2008. doi: 10.1109/TVCG.2008.165

[66] K. J. Worsley, J.-I. Chen, J. Lerch, and A. C. Evans. Comparing functional connectivity via thresholding correlations and singular value decomposition. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1457):913–920, 2005. doi: 10.1098/rstb.2005.1637

[67] S. N. Wright, P. Kochunov, F. Mut, M. Bergamino, K. M. Brown, J. C. Mazziotta, A. W. Toga, J. R. Cebral, and G. A. Ascoli. Digital reconstruction and morphometric analysis of human brain arterial vasculature from magnetic resonance angiography. *Neuroimage*, 82:170–181, 2013.

[68] X. Yang, L. Shi, M. Daianu, H. Tong, Q. Liu, and P. Thompson. Blockwise human brain network visual comparison using nodetrix representation. *IEEE transactions on visualization and computer graphics*, 23(1):181–190, 2016.

Evaluation of Glyph Design for Improved Probabilistic Categorization Accuracy

Leave Authors Anonymous **Leave Authors Anonymous** **Leave Authors Anonymous**

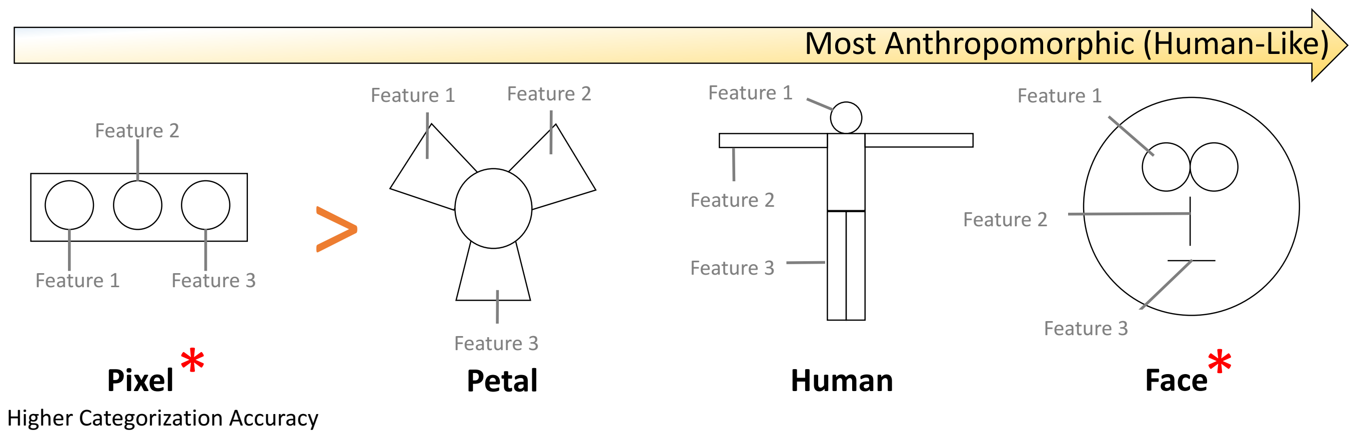


Figure 1. The glyphs evaluated in our study encode three variables as features. In the analogy of a doctor diagnosing a patient, each feature would encode the presence or absence of a symptom. In our study we found that the most abstract glyph ('Pixel') was significantly(*) more accurate than the 'Face' glyph.

ABSTRACT

Categorization tasks are common in everyday life, from sorting objects to a doctor diagnosing a patient's disease. In many cases, classification information is binary (e.g., patient's symptom present/absent) and visually represented. Suitable visual idioms for such multidimensional data are glyphs. To date, there has been no systematic evaluation of how to best encode probabilistic categorization data in glyphs. Here we ask: 'Whether visual data representation format affects categorization accuracy.' In our within-subject evaluation, 480 participants on Amazon's Mechanical Turk completed a probabilistic categorization task with two of four different glyph designs each of which encode 3 probabilistic features. The glyphs ranged from abstract to anthropomorphic. Based on existing literature in psychology and data visualization, we hypothesized that if visual representation affects accuracy than anthropomorphic glyphs would lead to higher categorization accuracy. However, subjects were significantly more accurate at categorization with the most abstract glyph design.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation: Miscellaneous
H.5.2.User Interfaces Evaluation/methodology

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI'16, May 07–12, 2016, San Jose, CA, USA

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 123-4567-24-567/08/06. . . $15.00

DOI: http://dx.doi.org/10.475/123_4

Author Keywords

Glyph, quantitative evaluation,
probabilistic categorization, memorability, anthropomorphic

INTRODUCTION

Categorization involves classifying objects based on their features. For example, sorting laundry before putting it in the wash by color and material. In many categorization tasks, the features of an object can occur in association with more than one category. For example, a physician who needs to diagnose whether a patient has a brain tumor must decide whether the patient's symptom of having a headache is due to a tumor or simply exhaustion.

Although human probabilistic categorization has been studied extensively [1, 2, 13, 14, 30], no studies to date have addressed the question of whether particular visual representations of relevant data features affect categorization accuracy. Visual encodings of this data, i.e., the presence and properties of attributes, are particularly useful for both data presentation (e.g., to present patient symptoms to a doctor in visual form) or education (e.g., train a doctor on how to accurately diagnose diseases). One visual encoding idiom that works particularly well with these data are glyphs, as there are typically multiple attributes to represent. Glyphs are visualizations well suited for multidimensional data in which dimensions are encoded to marks in the visual or pictorial representation.

To date, there has been no study or systematic evaluation of how to best encode probabilistic categorization data with glyphs. Which raises a question does the visual representation of the data affect categorization accuracy? And if so, how should the data be visually represented to maximize categorization accuracy? An accurate categorization is clearly important, especially in the scenario of a doctor diagnosing a patient. As categorization is learned through examples and experience, memorability comes into play as a key factor in correctly recalling which features or properties correspond to which category, i.e., categorization rule. Knowlton et. al. [20] empirically demonstrated that the use of declarative or explicit memory in addition to skill based learning of categorization rule leads to higher probabilistic categorization accuracy.

Consequently, for the study presented in this paper, we have two hypotheses with the first hypothesis as a validation and premise for our second hypothesis:

H1: There is an effect of glyph design on categorization accuracy.

H2: A memorable glyph design will aid in learning and recall resulting a higher categorization accuracy.

In this study, we base second hypothesis on the body of literature which deals specifically with memorability of graphical objects in the context of visualization and data encoding. Previous research has examined the memorability of natural images [19] as well as data visualizations[4]. This work has demonstrated that the inclusion of representations of people and human faces result in a significant improvement in memorability for natural images [19] as well data visualizations [4]. Research in visualization has also shown that pictorial data encodings are recalled more accurately than simple bar charts [15] when working memory is under heavy load. Consequently, to test our hypothesis that a memorable glyph representation would result in a higher categorization accuracy, we evaluated the effectiveness of anthropomorphic (human-like) glyphs as compared to abstract glyphs.

In order to evaluate the effect of glyph representation on categorization accuracy, including the effect of anthropomorphism, we conducted a within-subject study with 480 participants on Amazon's Mechanical Turk. Each participant completed a probabilistic categorization task with two of four different glyph designs each of which encode 3 probabilistic features. Two of the glyphs were of abstract design and two of the glyphs were human-like so that we could observe whether there was a positive benefit to the more memorable anthropomorphic glyphs. Contrary to our hypothesis(H2), we found participants were significantly more accurate with abstract than anthropomorphic glyphs. The ***Pixel glyph visual encoding generated the most accurate categorization performance and lead to statistically significantly higher accuracy than the Face glyph***. In addition, ***participants felt less confident with anthropomorphic glyphs*** in comparison to the abstract glyphs when performing the categorization task.

To gain further insight into these results of our study we conducted a post-hoc data analysis of the strategy data. Categorization strategy information was gathered and analyzed in two forms: *'qualitative' self-reported data* and *quantitative experimental data* from categorization task. The results of the strategy analysis demonstrate a correlation between categorization accuracy and visually salient features of the anthropomorphic glyphs. In this paper, we discuss the implications of our results for glyph design in real-world probabilistic categorization tasks as well as reflect on abstract and anthropomorphic data encoding.

Contributions: We present the first study to measure the effect of glyph design on probabilistic categorization and report results which demonstrate that glyph representation can affect task completion accuracy. To the best of our knowledge, this study is the first online probabilistic categorization experiment hosted and run on Amazon's Mechanical Turk platform [6]. Leading to a more diverse data sample compared to lab studies.

In the following sections we will review the concept of probabilistic categorization as well as relevant related work. We then detail our glyph design rationale and experimental design, and then conclude with a presentation and discussion of our results.

PROBABILISTIC CATEGORIZATION

In order to understand the purpose and content of our study, we first introduce the concept of probabilistic categorization. In a categorization task the available information is limited to a set of observations that are associated with their categories non-deterministically, i.e., the presence of a given criteria does not guarantee a particular cause, involving choices between indistinguishable possibilities. These non-deterministic categorizations are called **probabilistic categorization** tasks [10].

Experiment Design Justification: Traditional experiments in category learning [14, 20] use a two phase category learning experiment. The first phase is the "receive feedback" phase on patterns of features in which on a trial-by-trial basis the experiment trains the subject's criteria for choosing a given category. In the case of training a doctor for diagnoses, the study subject will learn to categorize a patient with a symptom of headache as a more likely case of exhaustion or dehydration rather than the lower probability but still possible category of rare brain infection. Over many trials we expect these criterion to settle around an unbiased point at which the subject is likely to choose a particular category for a pattern of features. The second phase of the experiment measures the effect of category learning in terms of accuracy. Therefore, in the second phase it is expected that the doctors are able to accurately diagnose patients based on the receive feedback phase. Our study is conceptually identical to the traditional experiments i.e. subjects first learn the probability of features associated with each category, and then the experiment measures the effect of category learning in terms of accuracy.

However the implementation of our study slightly differs from traditional experiments. The main differences exist in execution of the feedback and response phases. In traditional experiments, participants are asked to use an active learning method. For instance, participants are asked to guess the category without any training trials. Additionally, there is a time constraint and participants have to respond within this time frame. Also, in the traditional experiment participants continue to receive feedback for the entire task. In our experiment, we use a passive learning method, where participants are explicitly told that they will first be trained and then tested. During our testing phase participants do not get feedback about the true category. These conditions are more similar to many real-world categorization tasks. For complete experimental procedure refer to Sec. 5.1.

In this experiment we use the same feature combination for training and testing phase (Sec. 4.1) and employ a predefined set of patterns as was adopted by the study conducted by Gluck et. al. [14]. In the second experiment of the paper [14] it is evident that participants were trained on a predefined combination of features.

Memorability vs. Category Learning: Memory is essential for the successful execution of a categorization task, however the category learning paradigm goes well beyond simple memorability. Whether or not a subject makes decisions and receives feedback in the process of learning categories, subjects must remember what they saw on previous trials in order to complete the task. However when the relationship between categories and features is *probabilistic*, each feature can occur in each category. Simply recalling which feature belonged to which category is insufficient for accurate categorization - the subject must also infer the probability that a feature will occur in a given category and formulate a decision rule based on this inference.

RELATED WORKS

Data representation in categorization tasks: In category learning studies the use of visual features are common but their design is not justified. Some visual encodings are ad-hoc as in the weather prediction task [13, 14], where authors present each feature with a card that had a particular geometric pattern. Aron et. al. [2] used a potato head glyph (similar to the anthropomorphic glyphs in this paper) with features mapped on a hat, eyeglasses, mustache, and bow tie, and the glyph had no direct relation to the task. Other studies focused instead on the task and used abstract glyphs, e.g., Shepard et. al. [30] used shape, size and color to encode features. Their task was aimed to discover how variation across these channels affected classification accuracy. Alfonso-Reese et. al. [1] used bar charts as they had to represent continuous-valued features. In a human centered task like categorization, use of visual stimuli without knowledge of their perceptual effects, or whether visual encodings effect categorization task performance, may give rise to erroneous results. We aim to fill this gap by studying the effect of visual representation on categorization accuracy.

Strategy in Categorization Tasks: A categorization strategy is a rule people use to predict the category for an object. Categorization strategies are widely studied and often debated. While it is hard to cover all the strategies, we will discuss a recently published set of strategies by Gluck et. al. [14]. Gluck's strategies are relevant as they originate directly from subject responses in probabilistic categorization tasks (summarized in Fig. 2):

1. Multi-Cue Strategy: People perform inclusive categorization in which they use all the features to read the visual stimulus.

2. Singleton Strategy: People learn one 'primary' stimulus and based on this stimulus guess categories for other stimuli depending on how similar or different they are from the primary.

3. Single-Cue Strategy: People categorize on the basis of presence or absence of a single feature in the stimulus.

Gluck and others [24] also show that participants who used a Multi-Cue strategy had the highest categorization accuracy, and those who used the Single-Cue had the lowest accuracy [14]. In Section 7 we present a post-hoc strategy analysis to see if participants in our categorization task employ similar techniques.

Probability and Visualization: Previous research in the communication of probabilities [25, 26] and uncertainty [18, 27, 28] have outlined the challenges associated with representing probabilistic concepts. For example, in the paper by Micallef et. al. [25], the authors summarize that people are bad at making probabilistic inferences. These challenges also extend to critical jobs like medical

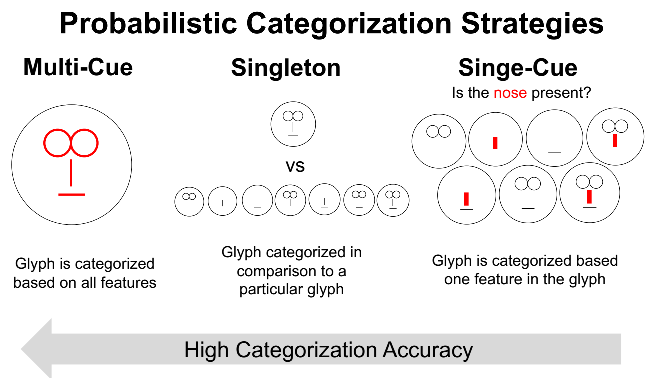


Figure 2. Three possible strategies which participants use while performing probabilistic categorization.The multi-cue strategy was found to be most efficient strategy and leads to highest accuracy, while single-cue strategy fares the worst and leads to lowest accuracies in the task.

diagnosis and judicial systems where results of incorrect probabilistic understanding can be catastrophic. Therefore, guiding research towards better visual representations to communicate probabilities can minimize the risk of such categorization failures. Similarly, if a visual representation helps people learn probabilities more effectively, it will directly impact categorization tasks in medical diagnosis and other domains that use categorization tasks.

Glyph Visualization: Glyphs are visual objects that are used to represent multidimensional datasets [3]. The glyph idiom maps one or more attributes of the data onto one or more of the visible marks of the object. An example of a glyph visualization is the Chernoff Face [8]. Chernoff used schematic faces to represent multidimensional data. Data was mapped using facial features eyes, nose, mouth, etc. The glyph design space is large, and their usability is widely studied [12]. Even with this large space, there are very few examples and studies which discuss how to encode binary data with glyphs. On previous study of particular relevance to this paper is [22] which evaluated glyphs with binary data [22]. Most of these past experiments were conducted with quantitative data which is different from the binary data in our study. In Section 4.2, we discuss and justify the need for binary data encodings in our study and motivate our glyph designs.

REPRESENTING MULTIDIMENSIONAL STIMULI AS GLYPHS

Multidimensional Stimuli

In probabilistic categorization a stimulus is made up of one or more features. Each feature maps to a dimension in our dataset. All stimuli can be visually encoded and represented by a glyph with multiple features as show in Fig. 1. In our analogy of a doctor diagnosing a patient, each feature represents a medical symptom of a patient, e.g., headache, blurred vision, etc., and the stimuli encodes which symptoms are present. A visual representation, e.g., glyph, can visually present to the doctor the symptoms present in the patient. In our study, all the features are of binary data type, i.e., they can take 0/1 value. Each feature can be present or absent from the stimulus based on the situation, therefore there are 2^n unique permutations of the feature in a stimulus. Bruner et. al. [5] describe every category as having two components: (1) Features that a person must look for in order to decide whether a stimulus belongs to one of the categories, and (2) a rule for which conjunction of constraints on the features will qualify as a positive instance of the category. Therefore, the stimulus representation for when no feature is present provides no information to categorize a concept and can be visually confusing, especially when people expect a visual cue to classify a stimulus. Consequently, the feature with no stimulus is removed allowing $2^n - 1$ total permutations. This convention of removing a stimulus with no evidence has also been demonstrated in earlier experiments [14, 20]. Our study uses three dimensional features, therefore each stimulus can have 7 permutations shown here as a data matrix:

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

S is the stimulus matrix, in which each row of the matrix is a permutation of features for the stimulus. Visual representation of these permutations are explained in the next section.

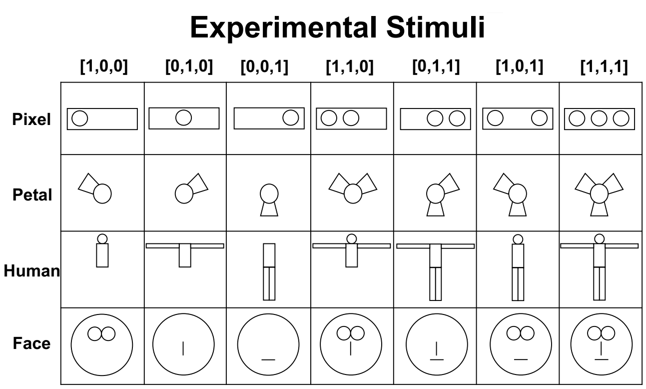


Figure 3. Mapping of data matrix *S* (columns) to marks in the glyphs (rows) (Sec. 4.1). Each feature is either present or absent from the glyph as the underlying data is binary. Each feature combination is called a *stimulus*. These stimuli are used in the categorization tasks to train and test probabilistic categorization.

Glyph Design

The glyphs used in this study were carefully and purposefully designed, discussed, and refined. In this section we motivate and explain our glyph design.

Abstract vs. Anthropomorphic: Anthropomorphism is the attribution of human traits or qualities to non-human entities. A 'Face' or 'Human' glyph (see Fig. 1) is anthropomorphic as their visual features are distinguishable and the glyph as a whole is a familiar human-like representation even if drawn in a non-realistic manner. In contrast, Abstract glyphs are visual forms with primitive shapes which have no associated intended meaning. In our study, the 'Pixel' and 'Petal' glyphs (see Fig. 1) are abstract. The design of these abstract glyphs ensures that there is no easy distinguishable or recognizable characteristics associated with visual identity of the individual features. To implement this definition, we use a symmetric design for the abstract glyphs. Symmetry ensures no visual feature gathers extra attention or looks distinguishable from the other features. For example, in the 'Pixel' glyph all of the features are identically shaped circles which are equidistant from each other.

Based on the design of these anthropomorphic and abstract glyphs we hypothesized a variation in their performance in a probabilistic categorization task. It has previously been shown that schematic faces are easy to recognize [17], distinguish [29], and anthropomorphic figures make a visualization more memorable [4]. In addition, probabilistic categorization studies have found that the use of declarative (explicit) memory leads to an improved probabilistic categorization accuracy [20]. This previous work and reasoning lead us to our ***hypothesis(H2):*** as anthropomorphic glyphs are visually more distinct and recognizable, a participant might employ a declarative strategy and achieve enhanced probabilistic categorization accuracy.

Encode Binary Data: The primary design criteria in this project was to encode binary data in a glyph visual encoding. The visual encoding maps the binary value (0 or 1) to the absence or presence of a feature (Fig. 3). This is common in medical diagnosis, where doctors use a simple binary conditional logic when dealing with symptoms [21]. The logic can look like: if a headache(symptom) then exhaustion(diagnosis). Anthropomorphic glyphs have a natural advantage for presenting binary data due to our familiarity with human-like figures ease of feature recognition. For example, we can intuitively spot a missing feature like an eye or nose even in a schematic figure. However, an abstract glyph is not configured to provide a cue to spot a missing feature. This frame of reference is an advantage of anthropomorphic glyphs as it provides a reference for missing features and in our study is anchored by the torso in the body-like 'Human' glyph, and the outline of the face in the 'Face' glyph. For continuity, the abstract glyph designs in our study also incorporate a frame of reference paradigm through symmetry in order to aid in identification of missing features. The abstract glyph designs in our study include the 'Pixel' glyph with a rectangle which encapsulates pixels as a frame of reference, and the 'Petal' glyph with a circular mid-portion of the petal as a frame of reference.

Pixel: The abstract Pixel glyph uses point (circle) marks which resemble three pixels. This glyph includes a rectangle as a frame of reference around the circles. These circles are placed equidistant from each other arranged horizontally linear. This design uses the position of the circle to encode data, i.e., one feature per circle. This design is scalable for higher dimensional data as we can add any number of circles inside a rectangle without any loss of generality.

Petal: The abstract Petal glyph utilizes a radial layout with shape marks ("petals") around a common center point. Our petal glyph is similar to the flower [7, 23] and star glyphs [11, 22], as both of them show features in a radial layout and use position and length to encode data on the individual marks. This glyph is scalable (i.e., can add more petals for more data dimensions) and it is also space efficient due to it's compact radial layout.

Human: The anthropomorphic human glyph is of a human body form in which each data dimension is encoded to a anatomical part. The three features in our study were mapped to the head, the arms, and the legs. The torso serves as a visual reference point, and it is visible in conjunction with all features. The Human glyph is hard to scale for higher dimensional datasets as there are a limited number of body parts. An advantage of the Human glyph is itâĂŹs visual saliency in which each data encoding mark is substantially different from one another.

Face: The anthropomorphic Face glyph is of a human facial form in which each data dimension is encoded to an anatomical facial feature. The three features in our study were mapped to the eyes, the nose, and the mouth. The Face glyph is also restricted in scalability by the number of features that can be mapped because a human face can have only a finite number of distinct features. Face glyphs are designed to be neutral, including in gender and emotion, and any impression a viewer has of these facets is coincidence.

Next, we discuss our experiment design and methods.

METHODS

Participants: A total of 480 study participants (mean age=35, and gender participation of 48.7% women and 51.3% men) were recruited through Amazon's Mechanical Turk. All participants were based in the United States of America and had a 95% or greater approval rating. Before starting the experiment, participants indicated their informed consent in accordance with the guidelines of our organization's IRB and our approved study protocol. The experiment was conducted in a single 15 minute session. Study participants were monetarily compensated and received $2.00 for their participation.

Stimuli: We use glyphs that encode three features and vary in design from from abstract to anthropomorphic (Fig 3).

Experimental Variables: In this experiment the *independent variable* is the glyph design. This independent variable can take 4 discrete values: Pixel, Petal, Human, and Face. Our *dependent variable* is the probabilistic categorization accuracy.

Factors: As categorization accuracy can easily be influenced by the participant's personal experiences and biases to mitigate the confounding effects on the categorization accuracy we have four factors (independent controlled variables) in our experiment: three-feature categorization objects, binary data, a context-free abstract categorization task, and a fixed probability structure for all participants.

1. Three-feature categorization objects: In our experiment we used three discrete features (Fig. 1) in order to make the task difficulty nontrivial as well as make the experiment duration and difficulty reasonable. The use of only two features would make the categorization task trivial, while four features would result in an an exponential increase in the number of stimuli that would need to be tested in the experiment. For example, the four feature scenario would entail testing 15 stimuli for each glyph, and the experiment would last four times longer (~1hour), the five feature scenario would require 31 stimuli.

2. Binary Features: Binary features can be interpreted as a threshold implemented on continuous values (e.g., when a doctor evaluates a medical symptom as being either "outside" or "within" normal limits). A person will form a foundational case for the probabilistic categorization task in this case; this premise is used as the basis for a large number of category experiments (e.g., [14, 20]). An additional advantage of binary features in an experimental design is the resulting finite set of discrete values which are easy to teach and test in a limited amount of time.

3. Abstract Categorization Task: An abstract categorization task, like those used in the present study, asks the participants to treat the features as generic features with no associated meaning. For example, in Fig. 1, all of the features are marked as 1, 2, or 3 and not, for example in the Face glyph, "eyes", "nose", and "mouth". We also provide a neutral premise and wording to the categorization task "Your task will be to look at each figure and decide which family it's from" instead of "Your task will be to look at symptoms and decide patient sick or healthy" which may introduce unknown biases, especially with the anthropomorphic glyphs.

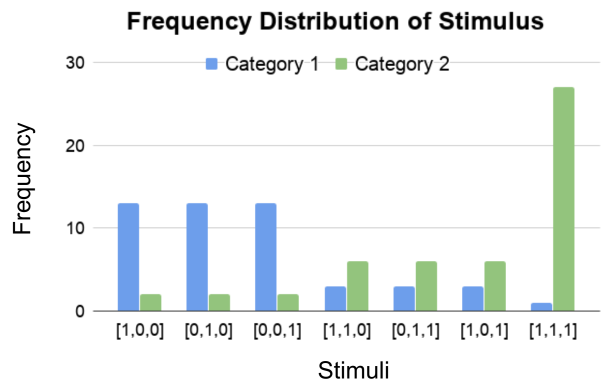


Figure 4. Probability distribution of stimuli 1-7 with frequency distribution in Category 1 and 2. The most likely category for stimulus 1, 2, and 3 is Category 1, and for stimulus 4, 5, 6, and 7 is Category 2.

4. Probability structure for categorization task: The probability structure, i.e., data source, of the experiment defines the relationship between features encoded in the glyph and their possible categories. As our task is probabilistic categorization, this means that each feature can appear in each category with some predetermined probability. Each feature has an independent probability of occurring in each category, and it is not affected by the presence or absence of other features. To combine the results across participants, each participant will observe the same probability distribution when they perform the task. In the current study, there are two candidate categories. Each category has an equal probability of occurring on any trial of the experiment, $p(C1) = p(C2) = 0.5$. Each feature has an independent probability of occurring in each category, that is: $p(F1|C1) = 0.8$, $p(F2|C1) = 0.8$, $p(F3|C1) = 0.8$, and $p(Fn|C2) = 1 - p(Fn|C1)$. The final probability values are obtained by calculating the conditional probability of observing a stimulus, from the stimulus matrix S (Sec. 4.1), given a particular category. Specifically, the probability that the m^{th} stimulus S_m will occur in C_i is given by:

$$p(S_m|C_i) = \frac{\prod_n p(S_{m,n}|C_i)}{\sum_m \prod_n p(S_{m,n}|C_i)}$$

$$p(S_{m,n}|C_i) = \begin{cases} p(F_n|C_i), S_{m,n} = 1 \\ 1 - p(F_n|C_i), S_{m,n} = 0 \end{cases}$$

Based on the probability structure, the frequency of each stimulus occurring in each category given N_{tot} total trials is given: $N_{S_m|C_i} = p(S_m|C_i)p(C_i)N_{tot}$. The frequency distribution for each stimulus with $N_{tot} = 100$ trials is presented in Fig. 4. *Justification of present probability assignment*: Assigning a probability of 0.8 to each feature makes the categorization non-trivial in terms of difficulty but challenging enough to elicit an observable effect on accuracy if one exists. Finally, this probability structure results in a straightforward decision rule that, if applied consistently, will maximize accuracy in the task. Namely, if the stimulus contains one feature only, it is most likely to belong to Category 1. If it contains more than one feature, it is most likely to belong to Category 2.

Between-subject Preliminary Study: In our between-subject study, each participant was only presented and evaluated with a single glyph representation. We collected data from 100 subjects on Amazon's Mechanical Turk across the four glyphs and the study duration was 8 to 10 minutes. Due to individual variances in categorization accuracy it was not possible to resolve effects in performance across the four glyph designs. A within-subject study design corrects for individual variances and, by having each individual categorize with different glyphs, allows us to measure the effect of glyph representation on categorization accuracy.

Within-Subject Study: Our final within-subject study design consists of six comparisons, in which each comparison measures the difference in categorization accuracy between two glyphs as shown in Table 1. Each comparison consisted of 4 experiments, and each experiment was further divided into 2 blocks. The structure of Exp. 1-4 was as follows:

Exp. 1: subjects trained and tested on Glyph "A" in Block 1 then Glyph "B" in Block 2.

Exp. 2: subjects trained and tested on Glyph "B" in Block 1 then Glyph "A" in Block 2. (Controls for the order effect)

Exp. 3: subjects trained and tested on Glyph "A"" in Block 1 then Glyph "A" in Block 2.

Exp. 4: subjects trained and tested on Glyph "B" in Block 1 then Glyph "B" in Block 2.

Exp. 3 and Exp. 4: Experiments 3 & 4 were designed to counter carryover effects in the evaluation, specifically learning and fatigue effects. A learning effect could positively affect the performance in the second task of this within-subject design. Due to this effect people could possibly learn the correct categorization strategy with the first glyph and subsequently improve with the second glyph. In contrast, fatigue could negatively affect a participant's performance. For example, fatigue occurs if a participant loses the motivation to perform the task well with the second glyph. Exp. 3 and Exp. 4 were designed to counter these learning and fatigue effects by measuring performance change when the task is performed with the same glyph in blocks 1 and 2 of the experiment. With this data we could accurately take these carryover effects into account in the final analysis.

Probability Structure in Blocks 1 & 2: As subjects could in theory learn the probability distribution of a stimulus occurring in each category from the first block and map it to the second block stimulus, we designed each block 1 and 2 to have different probability structures of equal difficulty to mediate this effect. To alter the probability distribution of Block 2, we flip the independent probability of each feature occurring in each category, that is: $p(F1|C1) = 0.2$, $p(F2|C1) = 0.2$, $p(F3|C1) = 0.2$, and $p(Fn|C2) = 1 - p(Fn|C1)$.

In each study (Table 1), Exp. 1 and Exp. 2 were each performed by 30 subjects. Exp. 3 and Exp. 4, which were designed to measure carryover effects, were conducted just once by 30 participants and reused in studies when required. For example, the Face-Face study was conducted once and then used in the analysis of the Pixel-Face, Petal-Face, and Face-Human studies.

NASA-TLX: As part of our within-subjects study design the participants were asked to complete a NASA-TLX survey form [16]. The survey gathers user feedback on seven task load measures for the categorization task: mental demand, physical demand, temporal demand, overall performance, frustration level, and effort. Per NASA-TLX convention, responses were recorded on a Likert scale of 1-7, where 1 stood for a positive response with labels **success** for performance and **low** for remaining measures. A measure of 7 stood for a negative response, with labels **failure** for performance and **high** for others.

Table 1. Within-Subject Study Design

Study	Exp. 1		Exp. 2		Exp. 3		Exp. 4	
	Block1	Block2	Block1	Block2	Block1	Block2	Block1	Block2
Pixel-Face	Pixel	Face	Face	Pixel	Pixel	Pixel	Face	Face
Pixel-Petal	Pixel	Petal	Petal	Pixel	Pixel	Pixel	Petal	Petal
Pixel-Human	Pixel	Human	Human	Pixel	Pixel	Pixel	Human	Human
Petal-Face	Petal	Face	Face	Petal	Petal	Petal	Face	Face
Petal-Human	Petal	Human	Human	Petal	Petal	Petal	Human	Human
Face-Human	Face	Human	Human	Face	Face	Face	Human	Human

Procedure

The entire study was conducted on Amazon's Mechanical Turk [6]. A custom JavaScript web application was developed to conduct the experiment. Once a worker completed the experiment, we ensured they were unable to participate again for repeat participation. In each experiment (Table 1) participants had to complete two categorization tasks. Each task consisted of the following steps:

1. Training Instructions: In the training block instructions, study participants were introduced to a glyph representation and told that they would learn how to assign each glyph to one of two families. Most importantly, participants were instructed A) that each glyph could occur in each family but would be more common in one family than the other, and B) that approximately half of the figures were from Family 1 and half from Family 2. Participants were instructed that they would be shown a set of glyphs with their respective family labels and that the participant should try to learn which glyphs occur in each family.

2. Training Phase: Participants then completed 100 trials of the training phase. On each trial, stimulus and category label were visible. The glyph was displayed with the true label (green number) and the screen remained visible until the subject advanced to the next trial.

3. Testing Instructions: This step consisted of a single screen instructing the participant that they would next view a series of stimuli (all of the same glyph encoding) and categorize them. Participants were asked to press on their keyboard the number "1" for Family 1 and "2" for Family 2 to pick the correct category. For the second part of the experiment, training instructions were slightly altered to notify participants that the probability model is different than with the first glyph. The instructions emphasized that this is a different task and participants have to pay equal attention to the training phase of this experiment.

4. Testing Phase: Subjects completed 100 trials of the testing phase. In the testing block the glyph was visible as in training however the correct categorization Family label was unmarked. In this way participants could not continue learning the probability distribution but rather allow us to evaluate how well they learned how to categorize features. On each trial we recorded participant response, true category of stimulus and the stimulus. After the testing block, participants completed a NASA-TLX based survey.

Demographic Survey and Qualitative Feedback: At the end of the experiment, participants were asked to voluntarily provide their age, gender, and any patterns they noticed or strategies they used to complete the task. They were also asked to report any issue they encountered in the experiment.

RESULTS

Response Bias: Response bias is the inclination of subjects to choose one category significantly over another. We believe participants who exhibited this trend in their data did not follow the instructions in the training block. We excluded such participant responses from our analysis to remove this bias. To calculate this bias in each experiment, subject responses from Block 1 and 2 were combined, and then a χ^2 test was used to determine whether responses were significantly ($p < 0.05$) biased towards one category or the other. These participants' data were excluded from further analyses. Out of 480 participants analyzed in the entire study (Table 1), we removed 150 participants.

Accuracy Analysis: Our experiment comprises 6 comparisons of categorization accuracy for each of the 4 glyphs listed in Table 1 under column 'Study'. Each of the four glyphs are comprised of 3 features and there are 7 stimuli for each glyph representing all possible combinations of features (see Fig. 3).

Each comparison consisted of 4 experiments, and each experiment contained 2 blocks: Exp. 1 tested glyph 'A' in block 1 followed by glyph 'B' in block 2; Exp. 2, glyph 'B' followed by glyph 'A'; Exp. 3, glyph 'A' followed by glyph 'A'; and Exp. 4, glyph 'B' followed by glyph 'B'.

We first calculate accuracy, α, for each of the s stimuli ($s = \{1,...,7\}$), in each block of each experiment in each of the c comparisons ($c = \{1,...,6\}$). Accuracy is calculated as the proportion of correct category responses for each stimulus, i.e., the number of trials on which the subject correctly categorized stimulus s divided by the total number of trials on which stimulus s was presented. For example, $\alpha^{AB}_{c,1,s,n}$ refers to the n^{th} subject's accuracy for stimulus s in the first block of the first experiment of comparison c in which glyph 'A' was tested, and $\alpha^{AB}_{c,2,s,n}$ refers to the accuracy for stimulus s in the second block of the first experiment of comparison c in which glyph 'B' was tested.

Next, we calculate each subject's difference in accuracy between each corresponding stimulus in the two blocks of each experiment. For example, the difference in accuracy for the corresponding stimulus s in blocks 1 and 2 of Experiment 1 for comparison c is given by:

$$\Delta\alpha^{AB}_{c,s,n} = \alpha^{AB}_{c,1,s,n} - \alpha^{AB}_{c,2,s,n}.$$

Next, we take the median of the differences across subjects for each stimulus in each experiment of each comparison. Then, in order to control for differences in accuracy due to fatigue and/or learning effects, we subtract the baseline median accuracy differences. The resulting across-subject median differences in accuracy per stimulus in Experiment 1 and Experiment 2 of comparison c are given by:

$$\Delta\alpha^{AB}_{c,s} = \text{median}\left(\Delta\boldsymbol{\alpha}^{AB}_{c,s}\right) - \text{median}\left(\Delta\boldsymbol{\alpha}^{AA}_{c,s}\right)$$

$$\Delta\alpha^{BA}_{c,s} = \text{median}\left(\Delta\boldsymbol{\alpha}^{BA}_{c,s}\right) - \text{median}\left(\Delta\boldsymbol{\alpha}^{BB}_{c,s}\right),$$

where $\Delta\boldsymbol{\alpha}_{c,s}$, in bold type, denotes the set of all subjects' accuracy differences for stimulus s in the respective experiment of comparison c.

Next we derive the observed difference in accuracy, per stimulus, for glyph 'A' in comparison c by averaging the differences from Experiment 1 and 2 (results in supplemental document):

$$\Delta\alpha^{A}_{c,s} = \frac{\Delta\alpha^{AB}_{c,s} - \Delta\alpha^{BA}_{c,s}}{2}.$$

Finally, we get the median observed difference in accuracy for glyph 'A' across stimuli for each comparison:

$$\Delta\alpha^{A}_{c} = \text{median}\left(\Delta\boldsymbol{\alpha}^{A}_{c}\right),$$

where, as above, $\Delta\boldsymbol{\alpha}_{c}$, in bold type, denotes the set of accuracy differences for all stimuli in comparison c. The results are shown in Fig. 5 and will be discussed in Sec 8.

Significance Test: In order to calculate the significance of the accuracy results presented in Fig. 5, as discussed below we derived the null hypothesis distribution for each glyph empirically through bootstrapping.

We assume each participant's number of correct responses for each stimulus in each block to be binomially distributed. Each distribution has parameters N_s equal to the number of trials on which stimulus s was presented, and p_s equal to the proportion of trials on which the subject responded correctly given stimulus s. We drew 10,000 samples at random from these distributions and divided each sampled count by its respective N_s resulting in 10,000 estimates of each subject's accuracy per stimulus per block. These estimates represent 10,000 simulated estimates of the results of our study from which we can construct 10,000 estimates of $\alpha_{c,[1\ 2],s,n}$ for each of the four experiments in each of the c comparisons. We carried out the analyses above on each of these estimates resulting in a set of 10,000 estimates of $\Delta\alpha^{A}_{c}$ which constitute the independent null distributions for each of the c comparisons.

In our analysis, we perform 6 comparisons, one for each study in the Table 1. To handle multiple comparisons appropriately, we use Bonferroni Correction method, i.e., we use adjusted values for computing the confidence intervals. For each comparison we use the following: $\alpha = .001$ and 99.9% confidence interval. If $\Delta\alpha_c^A = 0$ lies outside of the 99.9% CI, the observed value of $\Delta\alpha_c^A$ is significant at $\alpha = .001$. The CIs for each $\Delta\alpha_c^A$ are shown by the error bars in Fig. 5. Our results show that there is a measurable difference in categorization accuracy between glyph encodings, which supports our first hypothesis (H1). We also observe that ***the Pixel glyph has a significantly higher accuracy than the Face glyph*** which contradicts our second hypothesis (H2).

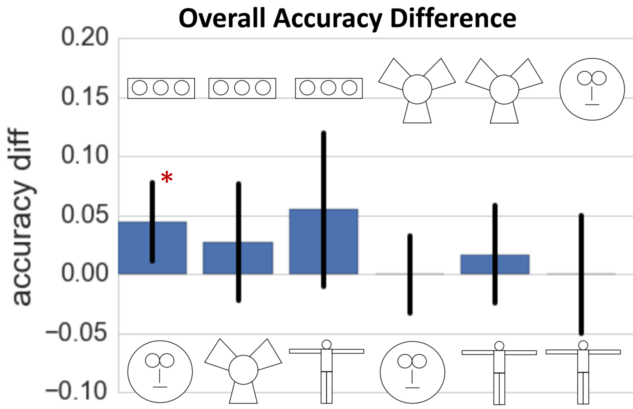


Figure 5. Summary of the average differences in accuracy between each pair of glyphs evaluated in the study. On the y-axis, positive ratios denote that glyphs on the top of the chart had greater accuracy, and visa versa for negative. For each glyph comparison, the 99.9% C.I. is plotted, and asterisks (*) denote Bonferroni-corrected significance in accuracy of one stimulus over other.

Table 2. Average subjective data responses to select NASA-TLX survey questions. The answers were rated on a 7-point Likert scale. Asterisks (*) indicate results of statistical significance.

#	Question	Pixel	Petal	Human	Face	Sig.?
1	How mentally demanding was the task? (1 = low, 7 = high)	4.09	4.08	**4.54**	4.26	*
2	How successful were you in accomplishing what you were asked to do? (1 = perfect, 7 = failure)	3.81	3.97	3.93	3.95	
3	How hard did you have to work to accomplish your level of performance? (1 = low, 7 = high)	4.56	4.42	4.77	4.77	
4	How insecure, discouraged, irritated, stressed, and annoyed were you? (1 = low, 7 = high)	3.36	3.19	3.42	**3.72**	*

Task Load Survey Results: From responses to our NASA-TLX (Sec. 5) survey questions we excluded responses to the questions of temporal effect and physical effort from analysis in our study due to their lack of relationship to our study's task. Results from the survey are summarized in Table 2. To test for statistical significance, we use a non-parametric Kruskal Wallis test. Significant results ***indicate that the Human glyph was significantly more mentally demanding*** ($p = 0.02$) , and ***participants felt insecure and discouraged using the anthropomorphic Face glyph*** ($p = 0.05$). The Face glyph overall received the most negative responses of all the glyphs designs.

POST-HOC CATEGORIZATION STRATEGY ANALYSIS

In this section we use self reported strategy data (Sec. 5.1, Demographic Survey and Qualitative Feedback) and quantitative categorization data (Sec. 5.1,Testing phase) to perform an exploratory strategy analysis. We use Gluck's strategy to explain the strategy differences (Sec. 3). We did not design the experiment to measure strategy difference, thus we consider the findings to be preliminary.

Self Reported Strategy Data: To analyze the self-reported qualitative data we reviewed responses for each glyph design. The responses suggest that people do not describe their strategies accurately. Roughly five out of ten strategies were either unclear or not meaningful. From the useful responses we observed that participants broadly followed the strategies described by Gluck. Here we list some sample responses for the Face glyph:

1. *Multi-Cue example*: "images that only had 1 feature usually were of one family while images that had 2 or 3 features were usually of another family" - The participant uses the optimal categorization strategy.

2. *Singleton example*: "mainly just full faces seemed to be more one than the other" - The participant appears to use the full face as a singleton object for classification.

3. *Single-cue example*: "I tried to remember which family associated more with full face and just eyes." - The participant appears to use a combination of singe-cue and singleton strategy.

4. *Ambigious strategy example*: "I just went with my gut."

Review of the responses we noticed the frequent use of the keywords "eye" and "head." Conversely, other features of anthropomorphic glyphs were rarely mentioned. To explore this further, a text-based search for keywords "eye" and "head" in the strategy data revealed a large number of references to this keyword. This explicit feature declaration supports the notion of a "Single-Cue" strategy. For example, one response stated: *"I thought that the heads belonged mainly to one family and same with the eyes."*. We list examples of additional Single-Cue strategy in the supplemental document (attached in Supplemental document). We believe this area of investigation into strategy analysis is a promising area of future research and investigation.

Strategy Derived from Quantitative Data: Our experimental results revealed that visual representations affect categorization accuracy and the Pixel glyph is significantly more accurate than the Face glyph. In this analysis we study strategy differences between the Pixel and Face glyphs with the categorization experiment responses. The conventional method to compare strategies is to compare optimal responses between strategies [14]. We compute the average percentage of optimal responses of participants from their testing phase data (Sec. 5.1). An optimal response percentage is the number of times a participant chose the optimal category for a stimulus divided by total number of trials. In Fig. 6, we show the average percentage of optimal response

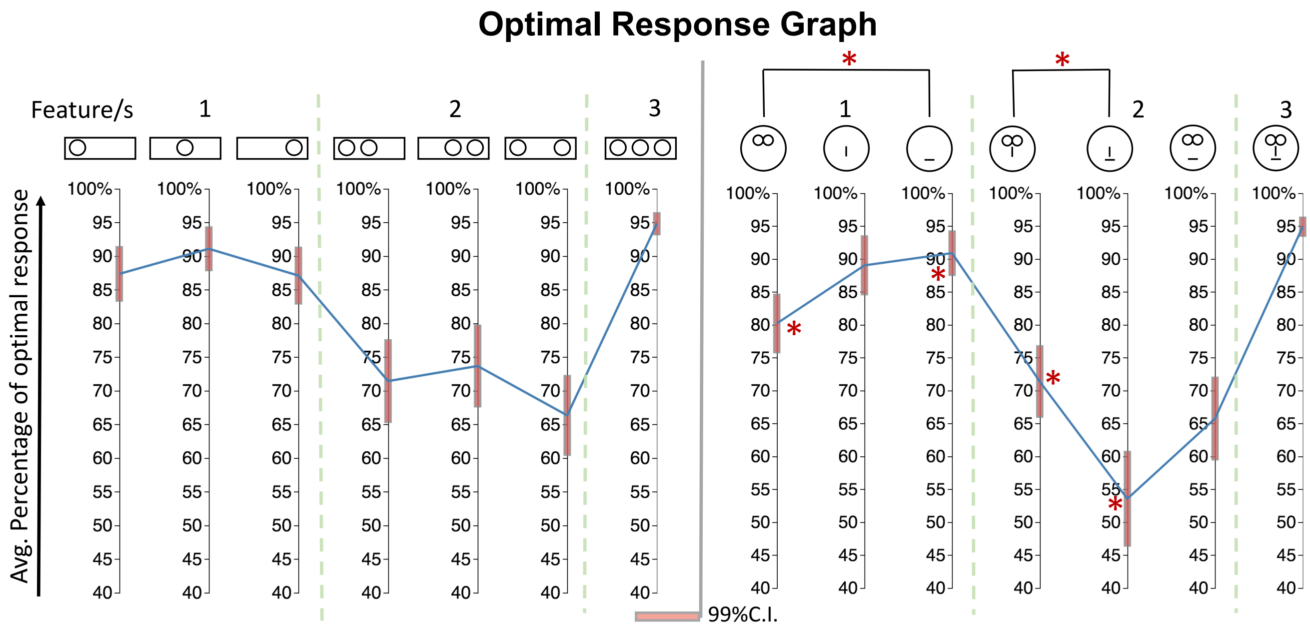


Figure 6. Average percentage of optimal responses, i.e., correct selection of the most likely category for a stimulus, for the Pixel and Face glyphs. The y-axis is superimposed with a red box which highlights the 99% CI of the distribution. Comparison among stimuli is within each of groups 1, 2, and 3 (i.e., stimuli in 1 feature group are not compared with stimuli in 2 features group). The groups are separated by green dashed lines. We found that the presence of the 'eye' feature affects the categorization strategy used by participants and accuracy. Significant differences are marked in the figures with an asterisk (*).

for each stimulus with the Pixel and Face glyphs. For the analysis we include data from all non-biased participants. Two significant differences are seen in Fig. 6:

1. *Face glyph with 1 Feature*: We observed that stimulus 1 with eye ([1,0,0]) has a significantly lower average optimal response percentage. The statistical significance for comparisons was calculated with a pairwise t-test which accounted for multiple comparisons (***p*<.001**) as compared to stimulus 3 with the mouth ([0,0,1]). This observation is unexpected as stimulus 1 (eye) and stimulus 3 (mouth) appeared in the training phase with the same probability structure (Fig. 4). There is no quantitative reason in our experimental design which can account for this difference, thus another factor (e.g., human prior for perceiving and interpreting faces) is responsible.

2. *Face glyph with 2 Features*: The proportion of optimal categorization for stimulus 4 with eye and nose ([1,1,0]) is significantly higher (***p*<.001**) than stimulus 5 ([0,1,1]) with no eyes. As with the previous observation, there is no explanation in our experimental design for this difference.

These observations indicate that face-like stimuli learned with the same probability in the training phase were associated with different categorization behavior in the testing phase. Based on the significant trend, we speculate that with the Face glyph a significant number of participants constructed a categorization rule, e.g., if eyes are present as a feature it belongs to one category and otherwise not (Single-Cue strategy). In contrast, with the Pixel glyph stimuli with the same learned probabilities there is no significant differences in the average optimal responses. These results suggest that people pay equal attention to all features and use Multi-Cue strategy for categorization with the Pixel glyph, but not the Face glyph. We discuss these results in more detail in the next Discussion section.

DISCUSSION

The results of our experiment support our first hypothesis that there is a difference in accuracy between glyph representations. However, our results do not support our second hypothesis that glyphs with an anthropomorphic design result in more accurate categorization. To the contrary, we found that the abstract Pixel glyph has a higher categorization accuracy than anthropomorphic Face glyph. Our experimental results are broken down into two different topics of discussion:

1. Differences in glyph design complicate comparisons across studies of categorization accuracy.

2. Some designs are associated with more accurate categorization performance than others.

Our results demonstrate that glyph design affects performance in probabilistic categorization. This result is important because previously conducted categorization experiments (Sec. 3) did not consider the design of visual stimuli could have an effect on results of their categorization test. In the light of our findings, we argue that designs like potato head glyph [2] or the cue-cards used in the weather prediction task [20] may have effected categorization performance. Therefore, the interpretation of categorization accuracy and strategies, or study of the effect of memory on data visualization, should take into account the possible effect of glyph design in the experiment. Our results indicate that the results obtained from previous studies might need further validation.

The second point of discussion is how the current study and future work could shape the understanding of visual stimuli for categorization tasks. In the present study, we ask "How should the data be visually represented to maximize categorization accuracy?" Our research explores a small but meaningful design space to understand the differences in categorization accuracy. Our results show that, contrary to our expectations based on published literature, abstract designs can have higher categorization accuracy than anthropomorphic designs. We justify this trend through analysis of the variation in categorization accuracy. In the sections below we discuss the evidence we found to support that this difference arises from strategy difference and discuss overall design takeaways.

From the results of our strategy analysis (Sec. 7), we speculate that participants use an optimal categorization strategy (Multi-Cue strategy) for the Pixel glyph and use a sub-optimal strategy (Single-Cue) for the Face glyph. Analysis of the optimal response data suggests that the eye feature is processed differently than the other elements of the Face glyph. We speculate that pre-attentive visual processing of human faces may explain this effect. Based on their social significance, the eyes have a pop-out impact which aids the pre-attentive processing [9], i.e., eyes attract higher attention than other features of the Face glyph. Consequently, eyes bias the categorization strategy of participants. In the self-reported feedback, we found similar patterns exist for the head feature in the Human glyph. Therefore, although we predicted that anthropomorphic glyphs would aid memory and guide participants to higher categorization accuracy, we instead see the opposite effect that our anthropomorphic glyphs negatively effect performance and lead to biased information selection and processing.

Based on the results from the post-hoc strategy analysis, we discuss two design recommendations for categorization tasks:

1. We propose that if the categorization task requires equal attention for all features, it is essential that glyph designers use an encoding in which all features are equally perceptually salient.

2. Alternatively, if the categorization task includes a subset of features that should be weighted more heavily in the observer's decision, such as in the doctor example indicating a symptom which is critical or deadly, then glyph designs with very salient or significant features (e.g., anthropomorphic designs) could be used to take advantage of human pre-attentive selection processes. However, because this method also affects how non-salient features are perceived, it is vital to be cautious of the consequences of using this encoding strategy.

NASA TLX Survey Feedback: As shown in Table 2, participants felt they had to work harder at the categorization task with the Human and felt more insecure with their task completion with the Face anthropomorphic glyphs. These increased cognitive load factors and insecurity may have contributed to the poorer categorization accuracy with anthropomorphic glyphs.

Towards Generalized Results: Our study takes an important step in the direction of general results for glyph encoding performance with categorization tasks including comparison of abstract glyphs with anthropomorphic glyphs. We found that participants perform the categorization task significantly worse with the most anthropomorphic (Face) glyph in our study. This results naturally leads to the inquire "Do other glyph encodings with anthropomorphic or natural features follow our results?" For example, if we use a dog-shaped glyph, will the dog head or face be processed differently than a human? What about a car-shaped glyph? These differences are hard to predict as there is significant variation in the participation strategy and may only be addressed with detailed future empirical studies. Nevertheless, based on the strategy feedback from our participants and categorization accuracy differences in our data, we believe that anthropomorphic glyphs in particular play a role in the categorization accuracy. In order to make broader statements we need a more comprehensive understanding of the relation between glyph design and categorization accuracy. We believe that a general framework may be built from a taxonomy that analyses glyphs based on different attributes like design specification, dimensionality, cultural context, and perceptual and attentional factors.

Table 3. Experimental Results Summary (Significant*)

Experimental Results for Probabilistic Categorization Task*
Glyph design affects categorization accuracy.
The most abstract glyph (Pixel) has significantly higher categorization accuracy than the most human-like (Face).
Participants lacked motivation with the Face glyph, and felt the task to be significantly more difficult with the Human glyph.
Strategy Analysis Results – Qualitative
The Eye and Head features of anthropomorphic glyphs play a role in the categorization strategy.
Categorization Strategy
For the Pixel glyph, people appear to employ a Multi-Cue strategy.
For the Face glyph, people appear to employ a Single-Cue strategy.*

CONCLUSION

Probabilistic categorization tasks are common in everyday life. We found that the visual encoding of probabilistic categorization data as glyphs for learning these probabilities and performing categorization tasks can affect human performance. In critical domains like medicine, higher categorization accuracy can potentially save lives. Even a small improvement in accuracy of a few percent could save a few lives out of a hundred. To find an effective data representation, we evaluated four glyph designs ranging from abstract to anthropomorphic. We hypothesized that glyphs which are more human-like and more memorable would lead to higher categorization accuracy. Contrary to our hypothesis, our results show that abstract glyphs lead to higher categorization accuracy. Our findings and results are summarized in Table 3. In contrast to our memorability hypothesis (H2), we learned that human-like objects might introduce biases as people relate differently to anatomically salient features which may hinder learning. Future and past categorization studies need, and real word applications which train people and implement decision support for probabilistic categorization, should carefully take into account their data representations.

REFERENCES

1. Leola A. Alfonso-Reese, F. Gregory Ashby, and David H. Brainard. 2002. What makes a categorization task difficult? *Perception & Psychophysics* 64, 4 (01 May 2002), 570–583. DOI:http://dx.doi.org/10.3758/BF03194727

2. Adam R Aron, Daphna Shohamy, Jill Clark, Catherine Myers, Mark A Gluck, and Russell A Poldrack. 2004. Human midbrain sensitivity to cognitive feedback and

uncertainty during classification learning. *Journal of neurophysiology* 92, 2 (2004), 1144–1152.

3. Rita Borgo, Johannes Kehrer, David H. S. Chung, Eamonn Maguire, Robert S. Laramee, Helwig Hauser, Matthew Ward, and Min Chen. 2013. Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. In *Eurographics 2013 - State of the Art Reports*, M. Sbert and L. Szirmay-Kalos (Eds.). The Eurographics Association. DOI: http://dx.doi.org/10.2312/conf/EG2013/stars/039-063

4. Michelle A. Borkin, Azalea A. Vo, Zoya Bylinskii, Phillip Isola, Shashank Sunkavalli, Aude Oliva, and Hanspeter Pfister. 2013. What Makes a Visualization Memorable? *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec. 2013), 2306–2315. DOI: http://dx.doi.org/10.1109/TVCG.2013.234

5. Jerome S Bruner, Jacqueline J Goodnow, and A George. 1956. Austin. A Study of Thinking, with an Appendix on Language, by Roger W. *Brown. NY: John Wiley & Sons* (1956).

6. Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. 2011. Amazon Mechanical Turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science* 6, 1 (2011), 3–5.

7. Michael Chau. 2011. Visualizing web search results using glyphs: Design and evaluation of a flower metaphor. *ACM Transactions on Management Information Systems (TMIS)* 2, 1 (2011), 2.

8. Herman Chernoff. 1973. The use of faces to represent points in k-dimensional space graphically. *J. Amer. Statist. Assoc.* 68, 342 (1973), 361–368.

9. Geen De Soete and Wilfried Do Corte. 1985. On the perceptual salience of features of Chernoff faces for representing multivariate data. *Applied Psychological Measurement* 9, 3 (1985), 275–280.

10. W. K. Estes. 1972. ELEMENTS AND PATTERNS IN DIAGNOSTIC DISCRIMINATION LEARNING*. *Transactions of the New York Academy of Sciences* 34, 1 Series II (1972), 84–95. DOI: http://dx.doi.org/10.1111/j.2164-0947.1972.tb02662.x

11. Johannes Fuchs, Petra Isenberg, Anastasia Bezerianos, Fabian Fischer, and Enrico Bertini. 2014. The influence of contour on similarity perception of star glyphs. *IEEE transactions on visualization and computer graphics* 20, 12 (2014), 2251–2260.

12. J. Fuchs, P. Isenberg, A. Bezerianos, and D. Keim. 2017. A Systematic Review of Experimental Studies on Data Glyphs. *IEEE Transactions on Visualization and Computer Graphics* 23, 7 (July 2017), 1863–1879. DOI: http://dx.doi.org/10.1109/TVCG.2016.2549018

13. Mark A Gluck and Gordon H Bower. 1988. From conditioning to category learning: An adaptive network model. *Journal of Experimental Psychology: General* 117, 3 (1988), 227.

14. Mark A Gluck, Daphna Shohamy, and Catherine Myers. 2002. How do people solve the "weather prediction" task?: Individual variability in strategies for probabilistic category learning. *Learning & Memory* 9, 6 (2002), 408–418.

15. Steve Haroz, Robert Kosara, and Steven L Franconeri. 2015. Isotype visualization: Working memory, performance, and engagement with pictographs. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. ACM, 1191–1200.

16. Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage Publications Sage CA: Los Angeles, CA, 904–908.

17. Donald Homa, Bram Haver, and Terrence Schwartz. 1976. Perceptibility of schematic face stimuli: Evidence for a perceptual Gestalt. *Memory & Cognition* 4, 2 (1976), 176–185.

18. J. Hullman, M. Kay, Y. S. Kim, and S. Shrestha. 2018. Imagining Replications: Graphical Prediction & Discrete Visualizations Improve Recall &; Estimation of Effect Uncertainty. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan 2018), 446–456. DOI: http://dx.doi.org/10.1109/TVCG.2017.2743898

19. Phillip Isola, Devi Parikh, Antonio Torralba, and Aude Oliva. 2011. Understanding the Intrinsic Memorability of Images. In *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2429–2437. http://papers.nips.cc/paper/4451-understanding-the-intrinsic-memorability-of-images.pdf

20. Barbara J Knowlton, Larry R Squire, and Mark A Gluck. 1994. Probabilistic classification learning in amnesia. *Learning & Memory* 1, 2 (1994), 106–120.

21. Robert S Ledley and Lee B Lusted. 1959. Reasoning foundations of medical diagnosis. *Science* 130, 3366 (1959), 9–21.

22. Michael D. Lee, Rachel E. Reilly, and Marcus E. Butavicius. 2003. An Empirical Evaluation of Chernoff Faces, Star Glyphs, and Spatial Visualizations for Binary Data. In *Proceedings of the Asia-Pacific Symposium on Information Visualisation - Volume 24 (APVis '03)*. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 1–10. http://dl.acm.org/citation.cfm?id=857080.857081

23. Yi-Na Li, Dong-Jin Li, and Kang Zhang. 2015. Metaphoric transfer effect in information visualization using glyphs. In *Proceedings of the 8th International Symposium on Visual Information Communication and Interaction*. ACM, 121–130.

24. Martijn Meeter, Catherine E Myers, Daphna Shohamy, Ramona O Hopkins, and Mark A Gluck. 2006. Strategies in probabilistic categorization: Results from a new way of analyzing performance. *Learning & Memory* 13, 2 (2006), 230–239.

25. Luana Micallef, Pierre Dragicevic, and Jean-Daniel Fekete. 2012. Assessing the Effect of Visualizations on Bayesian Reasoning Through Crowdsourcing. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (Dec. 2012), 2536–2545. DOI:http://dx.doi.org/10.1109/TVCG.2012.199

26. Alvitta Ottley, Evan M Peck, Lane T Harrison, Daniel Afergan, Caroline Ziemkiewicz, Holly A Taylor, Paul KJ Han, and Remco Chang. 2016. Improving Bayesian reasoning: the effects of phrasing, visualization, and spatial ability. *IEEE transactions on visualization and computer graphics* 22, 1 (2016), 529–538.

27. Alex T Pang, Craig M Wittenbrink, and Suresh K Lodha. 1997. Approaches to uncertainty visualization. *The Visual Computer* 13, 8 (1997), 370–390.

28. Kristin Potter, Paul Rosen, and Chris R. Johnson. 2012. From Quantification to Visualization: A Taxonomy of Uncertainty Visualization Approaches. In *Uncertainty Quantification in Scientific Computing*, Andrew M. Dienstfrey and Ronald F. Boisvert (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 226–249.

29. Noam Sagiv and Shlomo Bentin. 2001. Structural encoding of human and schematic faces: holistic and part-based processes. *Journal of cognitive neuroscience* 13, 7 (2001), 937–951.

30. Roger N Shepard, Carl I Hovland, and Herbert M Jenkins. 1961. Learning and memorization of classifications. *Psychological monographs: General and applied* 75, 13 (1961), 1.

Towards Identification and Mitigation of Task-Based Challenges in Comparative Visualization Studies

Aditeya Pandey*
Northeastern University

Uzma Haque Syeda†
Northeastern University

Michelle A. Borkin‡
Northeastern University

ABSTRACT

The effectiveness of a visualization technique is dependent on how well it supports the tasks or goals of an end-user. To measure the effectiveness of a visualization technique, researchers often use a comparative study design. In a comparative study, two or more visualization techniques are compared over a set of tasks and commonly measure human performance in terms of task accuracy and completion time. Despite the critical role of tasks in comparative studies, the current lack of guidance in existing literature on best practices for task selection and communication of research results in evaluation studies is problematic. In this work, we systematically identify and curate the task-based challenges of comparative studies by reviewing existing visualization literature on the topic. Furthermore, for each of the presented challenges we discuss the potential threats to validity for a comparative study. The challenges discussed in this paper are further backed by evidence identified in a detailed survey of comparative tree visualization studies. Finally, we recommend best practices from personal experience and the surveyed tree visualization studies to provide guidelines for other researchers to mitigate the challenges. The survey data and a free copy of the paper is available at `https://osf.io/g3btk/`

Index Terms: Human-centered computing—Visualization—Visualization theory,concepts and paradigms; Human-centered computing—Visualization—Visualization design and evaluation methods

1 INTRODUCTION

There are numerous ways to evaluate a visualization design or tool [16]. Out of the large variety of evaluation methods, one of the most common methods employs a comparative evaluation of visualization techniques to measure user performance, e.g., in terms of accuracy and time. Within visualization literature, these studies are identified as “comparative studies” [49] or “head-to-head comparisons” [37]. In visualization literature, comparative studies are used across a wide range of application domains to evaluate a variety of visualization techniques or encodings. For instance, Di Bartolomeo et al. [21] designed a comparative study to evaluate the effect of different timeline shapes on a subset of timeline tasks (Fig. 1). In another example, Plaisant et al. compared a novel SpaceTree [50] to a traditional node-link tree visualization to study the effect of their novel space-efficient tree visualization layout on a set of tree visualization tasks. A critical aspect of comparative study design is the selection of analytical user tasks that are ultimately used as a proxy to the real domain goals to evaluate the effectiveness of the visualization technique [49]. Given the critical nature of tasks, it is of utmost importance that researchers choose evaluation tasks that reflect the underlying research problem [43, 60].

*e-mail: pandey.ad@northeastern.edu
†e-mail:syeda.u@northeastern.edu
‡e-mail:m.borkin@northeastern.edu

While tasks play a pivotal role in comparative studies, existing visualization literature argues that the methods used for task selection and communication have several shortcomings. Plaisant [49] argues that the current process of selecting tasks for designing evaluation studies remains an adhoc process. The adhoc nature of the task selection process leads to problems like gathering the wrong task for evaluation [33] or gathering an incomplete set of tasks which do not cover the goals a visualization should support [33, 49, 55]. Another major task-based challenge is associated with the selection of a task abstraction technique. Task abstraction is the process of removing domain-specific terminology from the task description to promote easy understanding and adoption of the task-based results in application domains that are not directly related to the research problem [44]. Task abstraction is limited by ambiguity in choosing the correct abstraction framework [33, 48] and the method used to communicate the abstraction in the research article [33]. *The visualization community has recognized several task-based challenges in the context of evaluation studies, however these challenges have not been formalized in the context of comparative studies.*

Based on our personal experience with comparative studies (e.g., [11, 12, 21, 47, 59]) and the analysis of challenges from previous research articles [26,33,49,55] we identify four **task-based challenges** (C1-C4) that can directly affect the validity of comparative studies and thus influence the overall adoption and usability of the evaluation results:

- **C1:** Insufficient Justification of Task Source
- **C2:** Missing or Incomplete Task Abstraction
- **C3:** Inconsistent Task Description Format
- **C4:** Knowledge Gap in Task-Based Evaluations

In this paper, for each challenge we discuss its effect on the validity of different stages of a comparative study. Moreover, to investigate if these challenges exist in published academic literature, we survey tree visualization comparative studies and analyze 20 studies to identify if they are threatened by these challenges. Our analysis of the surveyed papers show that many studies have insufficient task justifications and have missing or incomplete task abstractions. Overall, we also found that task descriptions in the existing studies are inconsistent and researchers focus on a limited subset of analytical tasks within the possible task design space of tree visualizations. In the paper's challenges section (Sec. 4), we provide quantitative evidence for the existing challenges in the tree visualization studies. In addition to highlighting the challenges, we also offer practical recommendations to visualization researchers on how to identify the task-based challenges in their comparative studies and mitigate them.

Through this *survey* paper, we hope to draw the attention of the visualization community to a potentially problematic component of comparative studies. Through the BELIV workshop, we hope to initiate further discussion on this topic and work towards collectively identifying methods that can help our community resolve the task-based challenges in comparative evaluation studies.

Task: When did the earthquake happen?

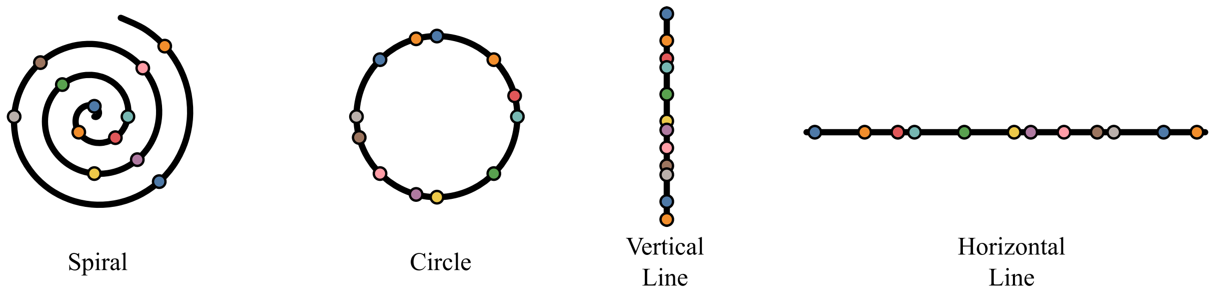


Figure 1: In a comparative study, Di Bartolomeo et al. [21] measured the effect of timeline shape on accuracy and time of user to find an event ("earthquake") on the timeline. The timelines shown in this figure are an abstract representation of the stimuli used in the original evaluation study.

Contributions: In this paper, we identify task-based challenges that exist in comparative studies through an analysis of existing visualization articles. We also provide evidence that existing problems manifest in published tree visualization studies by conducting a comprehensive survey of comparative studies for tree visualizations. Finally, we recommend guidelines that can assist researchers in mitigating the task-based challenges in their study.

2 Related Work

2.1 Review of Task-Based Challenges

Many researchers have raised issues and concerns within the information visualization community regarding how tasks are selected and communicated in the literature. In this section, we present an overview of the task-based challenges we found in the visualization literature.

Problems with existing task selection process: In a review of visualization evaluation challenges, Plaisant [49] highlights that the current method of task selection is ad-hoc, which can result in the collection of the wrong or incomplete tasks for an evaluation study. Saket et al. [55] also discuss the problem of limited tasks being used in existing visualization studies. In their review of existing evaluation studies, Saket et al. found that most papers only use a subset of tasks to benchmark the task-based effectiveness of basic information visualizations such as table, line chart, bar chart, scatterplot, and pie chart. They further conclude that due to the limited scope of the evaluation studies their results are hard to generalize for future work like curating task-based guidelines for visualizations. Kerracher & Kennedy [33] also discuss the challenges of task selection but in the context of task abstraction framework construction and validation. Their work identifies the threats associated with different task collection methods. For instance, if researchers use user interviews to collect tasks, they may encounter problems like identify the right people to interview or using the best interview method.

Problems with task description and explanation: Another challenge associated with tasks in information visualization is related to their description and explanation. In a review of information visualization evaluation challenges, Carpendale [16] argues that if visualization tasks are not defined clearly, they can be hard to test empirically and fail to provide insight into the usability of the visualization technique. In another study, Ziemkiewicz & Kosara [68] find that inconsistent phrasing of concepts in task descriptions lead to varying comparative results in the evaluation studies. Yet another problem with task description can be the lack of abstraction of the visualization tasks. Due to the lack of abstraction, Plaisant [49] reports that it is difficult to compare systems even with given tasks and datasets while discussing the main findings of the InfoVis contest [32] that was established to initiate and encourage evaluation benchmarks and methods. Since, comparative studies can span across different domains and are not tied to a single one, it becomes all the more important for tasks to be expressed in an abstract form using a standard language in order to enable comparisons across different domains.

To summarize, information visualization literature recognizes that task-based research presents several challenges. In this work, we systematically organize these task-based challenges for comparative studies.

2.2 Task Abstraction and Task Space

In information visualization literature, each domain has its own vocabulary for describing the data analysis goals or tasks [43]. The use of domain-specific task descriptions makes comparative studies hard to compare because of the task language variation. As a result of the task description variation, it is possible to miss out on common trends and patterns in the evaluation study results, which may help researchers formulate theory and knowledge regarding visualization use. To alleviate the problem of domain-specific language variation, researchers strongly advocate the use of task abstraction. According to Munzner, task abstraction can be defined as *"Transforming task descriptions from domain-specific language into abstract form allows you to reason about similarities and differences between them."* For example, the tasks "contrast the prognosis of patients who were intubated in the ICU more than one month after exposure to patients hospitalized within the first week," and "see if the results for the tissue samples treated with LL-37 match up with the ones without the peptide" may look different, but if expressed in abstract task terminology, both of the tasks are the same: "**compare** the value between two groups" [44].

Presenting tasks in an abstract and consistent manner or providing task abstractions along with domain specific tasks offer various advantages to visualization evaluations and design [33]. It allows evaluators to evaluate tools in comparative studies more efficiently [3, 5, 38, 58], enables better communication between researchers [53, 58], and most importantly provides a common vocabulary for analytical task descriptions [51]. There are many task abstraction frameworks (e.g., [4, 13, 44, 61]) but the level of granularity in task abstraction varies greatly within the visualization literature [13]. For example, Amar et al. [4] provides ten low-level analytic tasks that encompass a user's activity while using visualization systems in order to understand data. Shneiderman [61] presents a task taxonomy on seven tasks based on seven different data types. In contrast, Brehmer & Munzner [13] offers a multi-level task typology that differentiates why and how a task is carried out and what are its inputs and outputs. The typology breaks down visualization tasks into abstract and interdependent high-, mid-, and low-level actions that the user performs to carry out a specific task.

Although task abstraction is necessary to facilitate fair comparisons across different domains, it can only be done once the domain tasks to be performed are selected or generated. In order to do so, it is important to consider the task space, i.e., the list of all possible tasks that can be performed using the visualization tool(s). Schulz et al. [58] defines a design space that distinguishes five aspects of visualization tasks and consolidates the vast amount of task taxonomies, classifications, and frameworks that are scattered across the visualization literature in the forms of lists, descriptions, mathematical task models, domain-specific models of tasks, and workflows derived from task procedures. The task design space defines 5 dimensions: i) why a task is performed, ii) what type of data does the task need, iii) who is performing the task, iv) when is the task performed and in what order and, v) how is the task performed in terms of actions. The authors also put their design space for task into practice by applying it in climate impact research. Amongst some of the practices of task generation are deriving from literature [66, 67] or the author's own knowledge, interviews with domain experts [14, 21, 41, 52, 57] and reviewing existing systems for the tasks they support [17, 25]. For a full list of the task generation techniques and the threats they pose, please refer to Schulz et al. [58].

We have personally experienced the drawbacks of the above mentioned ways of task generation in some of our previous works. Extracting tasks from literature and existing systems is hard because of the lack of task abstraction and consistency in description. Also relying on the author's knowledge greatly increases the risk of missing out on important tasks, specially in the case of comparative evaluations where multiple domains can be at play. We adopted the method of interviewing domain experts in our previous work [11, 12, 21, 47, 59] and while this is an improvement from not justifying and validating the tasks at all, it also has drawbacks. For example, the domain expert's availability might be limited [23, 54]. Also, interviewing an expert from a single domain might run the risk of introducing bias in selecting representative tasks [34]. This is because while human-computer interaction (HCI) studies that involve usability testing of a tool with experts as participants indicate that three to five evaluators will suffice for usability testings depending on the particular problem [46, 62], there is no agreed upon rule on how many expert interviews is sufficient for obtaining a representative task list. This is an interesting research question that needs further investigation.

2.3 Comparative Evaluations in Visualizations

Data visualizations are created based on varying goals and strategies and hence the type of evaluation in each case would be expected to vary as per the aim of the researchers. Although controlled experiments and usability studies are the cornerstones of visualization evaluation [49], there are many other diverse metrics by which evaluations have been classified. Andrews [6], Ellis et al. [22], and Hilbert et al. [27] classify evaluations based on research goals such as summarizing the efficacy of an interface (summative), providing design guidelines (formative), comparing design alternatives (predictive), and understanding user behaviour, performance, thoughts, and experiences (observational and participative). Some make the distinction based on whether the data collected is qualitative, quantitative, or mixed and whether it is collected empirically or analytically [8, 19]. Researchers also differentiate evaluation methodologies based on research strategies [40], methods [31], and evaluation scope [18]. Munzner [43] breaks down evaluations based on the corresponding design stages of visualization development. Plaisant [49] surveyed approximately 50 information visualization user studies and derives four different high-level themes of evaluation: i) Comparison of design elements through controlled experiments [1, 11, 28], ii) usability evaluation of a visualization tool [15, 64], iii) controlled experiments to compare two or more tools [50], and iv) case studies of tools in realistic settings [65]. In our work, we focus on the thematic classification by Plaisant [49], specifically on themes (i) and (iii) pertaining to comparative evaluations between visualization tools or elements of design within different visualization tools or techniques.

We narrow down our focus to comparative visualization studies because the scope of task-based challenges in visualization evaluation is vast and beyond the scope of this paper. Moreover, comparative studies are one of the most common forms of evaluation methodology in visualization literature. In Lam et al.'s [37] meta analysis of 850 papers, they found that investigating user performances (UP) and user experiences (UE) were the most common themes of evaluations. Isenberg et al. also reported UE and UP to be the dominating evaluation scenarios [30]. Comparative evaluations can encompass all of these scenarios, and are mainly used to measure UP and UE [2, 11, 29, 42]. Finally, the tasks of comparative studies can span across different domains thus task abstraction is of paramount importance for study comparison. Therefore the focused systematic review of task-based challenges in comparative visualization studies in this paper is particularly important in order to make the comparisons fair, generalizable, and independent of domain expertise.

3 Identification of Challenges and Tree Evaluation Survey Methodology

The task-based challenges discussed in Sec. 4 were collected through a grounded theory approach [19]. The first author reviewed the visualization literature discussed in Sec. 2.1 and developed a list of task-based challenges. After this, all the authors examined the challenges and, with the backdrop of their personal experience and observations while designing comparative studies, evaluated the validity of the challenges. After a few rounds of iterations, all the authors agreed upon the four challenges discussed in Sec. 4.

As the challenges were gathered through a qualitative research method and based on the authors' experience, we found it extremely important to validate the challenges with existing comparative studies. Comparative studies are a ubiquitous evaluation method in visualization literature. Therefore an exhaustive survey of all published comparative studies was beyond the scope of this work. However, if we scope the selection of a study belonging to a particular data type, then we realized that the survey is feasible and within the scope of this paper. Since all the authors had prior research experience with tree visualizations, we chose to survey comparative studies in the tree visualization literature.

Survey Methodology: We searched for tree visualization comparative studies in the three most common digital libraries that publish the visualization research papers: IEEE Xplore, ACM Digital Library, and Eurographics Digital Library. In addition to the digital libraries, we searched for articles on Google Scholar because there is a chance that the digital libraries may not contain articles that may be relevant for the survey. On the IEEE Xplore library, we searched for the term: "Tree Visualization Evaluation". Further, we were able to filter articles only published at visualization conferences and journals, including IEEE Transactions on Visualization and Computer Graphics (TVCG), IEEE VIS, IEEE VAST, IEEE Pacific Vis, and Information Visualization(SAGE Journals). After applying the filter we found **30** papers on the IEEE Xplore library. Unlike IEEE Xplore, ACM Digital Library and Eurographics Digital Library do not have a feature to select the conference and journals. Therefore, we only selected papers with all the terms "tree", "visualization," and "evaluation" in the title. On applying these filters, we found **1** paper on ACM Digital Library and **1** paper on Eurographics Digital Library. On Google Scholar, the search term returned 770,000 results. In Scholar, the results are presented in order of relevance to the search criteria. We skimmed the first seven pages of search results (each page returns 10 papers) and determined that the first two pages had 50% relevant papers to our survey, but by the seventh page, most articles were irrelevant. However, for consistency across search phrases, with chose to include the first **100** papers (10

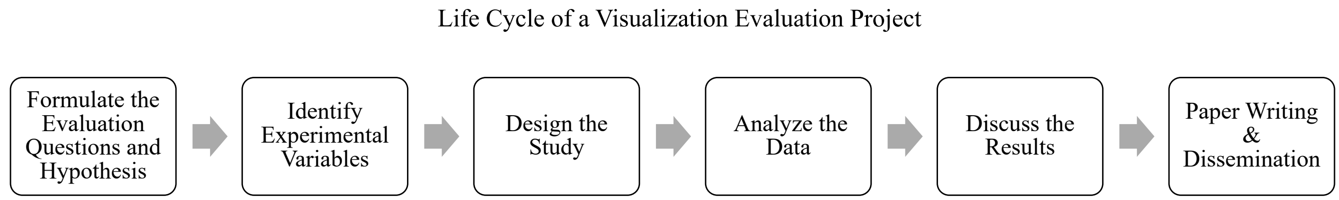


Figure 2: This figure presents an overview of an evaluation project. The evaluation project begins with research question selection, progresses with evaluation design steps, and deals with the collection and analysis of data and terminates with researchers presenting the results in the form of research articles. We use this figure to enable an effective discussion of the challenges in the context of the different steps a researcher goes through to conduct and publish an evaluation.

pages of returned search results) to ensure we captured all relevant material. From a corpus of **132** articles, we skimmed and filtered all the articles that were not comparative evaluation studies. In the end, we had a total of **20** articles that met our search criteria, i.e., they were comparative evaluation studies of tree visualizations. We provide the final list of the articles in the Supplemental Material.

4 TASK-BASED CHALLENGES

Based on our analysis of the existing visualization literature and our prior research experience, we found four task-based challenges associated with comparative studies. Each challenge poses a threat to the validity and usability of visualization evaluation results. In this section, we expand on the challenges to promote awareness about these challenges to the community and offer concrete recommendations to mitigate the challenges. To ensure that we use a consistent clear method to communicate the challenges, we introduce a "challenge reporting" template:

Challenge Reporting Template:

Description: A subjective explanation of the problem.

Cause: Identification of the probable factors that contribute to the challenge. Identification of causes is essential because it directly affects how we build recommendations or guidelines to mitigate the problem.

Effect: Identification of the steps in the life cycle of an evaluation project (Fig. 2) that may be affected by this challenge and discuss them in detail. This effect discussion is important as each challenge has the potential to invalidate an evaluation project.

Evidence: Presentation of evidence from the surveyed tree visualization comparative studies to support the challenge. For an objective analysis of the problem, we quantify the evidence using measures that can generalize to future surveys.

Guidelines: A proposed set of guidelines or best practices corresponding to each challenge that we argue may help mitigate the effect of the problem. These guidelines were extrapolated from our review of related works. We divide the guidelines into two categories: "Researchers" and "Community" based on the intended target audience for the guideline.

4.1 C1: Insufficient Justification of Task Source

Description: Comparative studies may fail to describe the source of the tasks and provide an explicit rationale for selecting the tasks. The tasks used in a comparative study represent the type of questions or queries researchers want the user to answer with a visualization. For instance, through the task "Which directory includes a deeper hierarchy: 'Flutes' or 'Guitars'?" [35] the researchers are trying to compare four tree visualization techniques on their capability to display the depth or height of a tree dataset. The tasks used in these studies are critical to the design of a study because an incorrect selection of tasks may discard the entire evaluation. Given the essential nature of tasks, we assumed that researchers would explicitly discuss the source and rationale of the included tasks in the research articles they write to summarize the experimental procedure and results. However, for the tree papers we reviewed, we found a number of papers (discussed later in the Evidence section) that did not have an explicit rationale for the tasks included in their study. Some papers did not even have a clear description of the source of the tasks.

Cause: Existing literature effectively guides how to report experimental methods and results of an evaluation study [20, 24], but they do not focus on the specifics of how to communicate the source and rationale of tasks used in a study. The primary cause of this problem may be a lack of a clear understanding of what constitutes a source and rationale for the tasks. In the guidelines section of this challenge, we present a list of task sources and provide some role model examples of how to describe the rationale for choosing a set of tasks.

Effect: Insufficient justification of the tasks can cause a problem in the very early stages of an evaluation project. If the researcher does not provide the fundamental reason for the task source, then all the steps after **Design the Study** in Fig. 2 become irrelevant because other researchers will not accept or use the results. Therefore, this challenge may have the most severe consequence and should be dealt with appropriately.

Evidence: Compared to the other parts of the methodology section like the evaluation procedure and data analysis, we noticed that task source and rationale were under specified in the reviewed tree visualization papers. Out of the 20 tree visualization papers, 50% did not mention a clear source for their tasks, and a staggering 70% of these did not provide a clear rationale behind the selection of tasks. For instance, Kosba et al. [35] describe the source as *"The task selection was also informed by a very early version of the InfoVis 2003 contest tasks. In some cases, questions had to be rephrased using a more technical terminology in order to make them unambiguous."* However, the paper did not discuss aspects such as "How did they rephrase the tasks?" and "What ambiguity made them to do it?" In another example, Barlow et al. [9] mention that *"The tasks used to evaluate the compact views were based on the requirements of the user in a data mining context."* However, they do not discuss the users nor do they reveal the methodology used for collection of the tasks. These examples demonstrate how the task source is under specified and how many tasks are included in the studies without proper rationale. We provide more details about other papers in the Supplemental Material.

Guidelines for Researchers: To mitigate this challenge, we recommend researchers mention and describe the source of the task explicitly in their published papers. Kerracher & Kennedy [33] identify seven categories of task sources: derive from literature, interview with domain experts, a survey of visualization experts, observational strategies, system review, author's knowledge, and derive from existing frameworks. All the above categories should be considered acceptable as long as researchers are clear about the motivation and reason behind choosing one source over the other. In addition to the source, researchers should also provide the rationale for selecting the set of tasks. The rationale is usually a justification that clarifies and supports the decision to choose a set of tasks. For instance, Santos et al. based their task selection on the fact that their visualization was designed to evaluate 3D information. The task selection was expected to test the tree visualization techniques that support 3D information display [56].

Guidelines for Community: Identifying tasks for evaluations can be challenging for researchers due to the lack of domain or field experts to interview, or lack of understanding of the literature to correctly identify analytical tasks through survey methods [33]. To resolve this problem, Plaisant [49] proposes the creation of **task datasets**. The task dataset is an exhaustive collection of visualization tasks built by a literature survey or collaborative data collection methods. Such datasets can be used as a verified resource for researchers to recognize tasks for a visualization evaluation. In addition to task dataset, visualization community also needs to develop broader guidelines to support visualization researchers to choose evaluation tasks. As discussed in "C1: Guidelines for Researchers", there are numerous methods for collecting tasks through expert interviews or survey of existing literature and taxonomies. However, we observed in our personal experience [21] that there is a lack of guidelines on when and how to use the resources. This challenge highlights that more work is required from the visualization community to build knowledge on methods to collect task for evaluation studies.

4.2 C2: Missing or Incomplete Task Abstraction

Description: Comparative studies may fail to abstract a visualization task that includes domain-specific terminology. In the tree visualization survey, we observed comparative studies often use domain-specific task-descriptions for the evaluation. For instance, **T1**: "Compare the directories '/projects/ravon/control/' and '/projects/ravon/navigator/'. Which one is the larger one?" [45] and **T2**: "Which directory has more direct sub-directories: '/hcil/about' or '/hcil/eosdis'?" [7]. T1 and T2 were used in two different studies, therefore the language used in the tasks is different. As discussed in Sec. 2.2 the usage of domain-specific task terminology can often be misleading, and readers may fail to notice a similarity between two tasks. As a result, visualization theory strongly advocates that tasks descriptions should be translated from domain-specific language into abstract form. We argue that the missing or incomplete task abstract reduces the transparency of a task used in the evaluation study and leaves the responsibility of abstraction on the person who is reading the paper in the future.

Cause: Information visualization provides many theoretical task abstraction frameworks but lacks guidance on how to use these frameworks in practice (see Sec. 2.2). The availability of several competing frameworks with limited guidance on how to use them makes it hard for researchers to choose the right abstraction method. Moreover, the literature also lacks clear consensus on the task abstraction specificity. For instance, in Fig. 3 (1) we can notice that authors provide a group level and task level abstraction in their study, i.e., for the first "Overview" is the group-level abstraction and "Deepest Subdirectory" is the task-level abstraction. However Fig. 3 (3) provides only a group-level abstraction without "Evaluate downward navigation extension". The combined effect of multiple abstraction frameworks and lack of guidance on how and what to abstract may be the main reasons for this challenge.

Effect: Missing or incomplete task abstraction can affect an evaluation project at multiple stages of Fig. 2. At **Design the Study** stage, without an abstraction researchers may use redundant identical tasks in the experiment. Abstraction will ensure that researchers add a variety of tasks to the evaluation. At **Analyze the Data** stage the lack of abstraction may miss the opportunity to analyze trends in the results. For instance, in tree visualizations, two common types of targets are "Topology" and "Attribute" [44]. If the researchers know the task abstraction, then they can analyze trends between topology and attribute tasks. For example, a Node-Link encoding for tree visualization is more efficient with topology tasks, but a Treemap is more efficient with attribute tasks. At **Paper Writing and Dissemination** stage without the task abstraction, other researchers may fail to analyze the results and reduce the usability of the evaluation results.

Evidence: For each tree visualization paper we surveyed, we reviewed the paper in detail and categorized it into one of three categories of abstraction:

- No Task Abstraction: The paper does not include abstract information with the tasks.
- Group-Level Task Abstraction: The paper does not abstract all tasks, but provided a group-level abstraction.
- Individual Task Abstraction: The paper contains abstractions for each task in the study.

We present the categorization results in Fig. 4 (Abstraction Distribution). The results demonstrate that 30% of the papers were missing task abstraction, while 30% only provided partial group-level task abstraction.

Guidelines for Researchers: To mitigate the challenges, we recommend researchers to adopt a suitable task abstraction framework. There is a lack of conclusive proof in the existing visualization literature that will support one abstraction framework over the other. Therefore, we argue the choice of abstraction framework should rest with the researcher. However, the researcher should justify the reason for adopting the particular task abstraction framework. Additionally, we argue that researchers should abstract each task they use in the comparative study if the task's terminology consists of domain or dataset-specific references. We believe task level abstraction will not just assist other researchers in understanding the tasks but also prove to help eliminate redundant tasks in the study design. Furthermore, task abstractions may also help with the outreach of the paper and increase its overall impact as researchers from diverse backgrounds can learn and benefit from the research.

Guidelines for Community: This challenge also raises a broader question on the usage and evaluation of extant task classification frameworks. Kerracher & Kennedy [33] summarize that the adoption, evolution, and demise of task classifications "in the wild" may provide valuable information about their descriptive abilities, comprehensiveness, usefulness, and usability. The visualization community needs to conduct such "in the wild" studies and develop guidelines that will assist researchers in choosing the right task abstraction frameworks to use under different conditions.

4.3 C3: Inconsistent Task Description Format

Description: Comparative studies do not have a standard method for how to frame the task used in an evaluation study which leads to inconsistent task descriptions across studies. In our survey of comparative evaluations of tree visualizations, all papers contain an explicit list of tasks in the body of the paper. We found that research papers often use more than one task to evaluate the visualization's performance in terms of task accuracy and completion times. However, we noticed that the method of task presentation was inconsistent across papers. Tasks descriptions used in three different papers are shown in Fig. 3 (1-3). In Fig. 3 (1), the authors also present the abstract categorization of the task and the instructions they presented to the participants in the study [7]. However, in Fig. 3 (2 & 3) the authors present only the task that they used [10, 56]. In a published paper the task description and experimental instructions are important to ensure that other researchers can replicate the study [24]. The lack of specificity and supplemental information about the tasks used in the comparative study may inhibit the study's accurate replication.

Cause: The inconsistency in task description may be attributed to the lack of proper information on task-reporting best practices within

1

A1	Overview	Deepest Subdirectory	Find the deepest subdirectory inside the directory "pad++" (/hcil/pad++). Write the name of this directory into the answer field to the right and then press "Continue...".
A2	Overview	Most Subdirectories	Find the directory inside "ndl" (/hcil/ndl) with the most direct subdirectories. Write the name of this directory into the answer field to the right and then press "Continue...".
A3	Search	Find Directory	Find the directory "yidemo" (/hcil/lifelines/yidemo). When you have found the directory, write "OK" or "found" into the answer field to the right and then press "Continue...".
A4	Search	Find File	Find the file /hcil/treemaps/treemap2000/images/banner-logo-large.gif. When you have found the file, write "OK" or "found" into the answer field to the right and then press "Continue...".
A5	Count	Count Subdirectories	Count the number of subdirectories directly inside the directory "/hcil/pubs". Write the answer into the answer field to the right and then press "Continue...".
A6	Count	Count Files	Count the number of files directly inside the directory "/hcil/qp". Write the answer into the answer field to the right and then press "Continue...".
A7	Compare	Compare Subdirectories	Which directory has more direct subdirectories: "/hcil/about" or "/hcil/eosdis" ? Write the answer into the answer field to the right and then press "Continue...".
A8	Compare	Compare Files	Which directory has more files directly inside: "/hcil/spotfire" or "/hcil/spacetree" ? Write the answer into the answer field to the right and then press "Continue...".

TDAI*: A Comparative Study of Four Hierarchy Browsers using the Hierarchical Visualisation Testing Environment (HVTE) by Andrews et al.*

2

2 Practice Task
a) Locate a directory given a path and select a file with a given size (theme: "fruit").
b) Click on the home button (returning to the root).
c) Return to the "fruit" directory in a) and select a file with a given name.
3 Locate and select the smallest file of the type "PDF".
4 Same as 2 but with a different target directory. (theme: "architecture")
5 Same as 2 but with a different target directory. (theme: "animals")
6 Locate and select a file given a path.
7 Return to the directory from task 5 with the animal theme.
8 Select the directory containing the only file of a certain type.

TD*: The effect of animated transitions on user navigation in 3D tree-maps by Bladh et al.*

3

extensions to original technique	
Task 1	Identify all individuals of a certain generation (all great-grand parents)
Task 2	Identify a specific individual, given the relation pertaining to the central individual
Task 3	Identify the relation that a specific individual pertains to the central individual
Evaluate downward navigation extension	
Task 4	Identify the children of a given individual
Task 5	Identify siblings of a given individual
Task 6	Identify a remote descendant (8 generations) given the relevant lineage

TDA*: Extending the H-Tree Layout Pedigree: An Evaluation by Santos et al.*

Figure 3: The figure shows variation in task description format used in tree visualization comparative studies. This figure highlights that there is a lack of general technique to report the tasks used in comparative studies. The most specific description is shown in the top figure (1), where the tasks are presented with abstraction and instructions [7] (TDAI). The task descriptions shown in bottom row (2 & 3) do not have instructions for the tasks [10, 56] (TD), that may hinder its replication by other researchers.

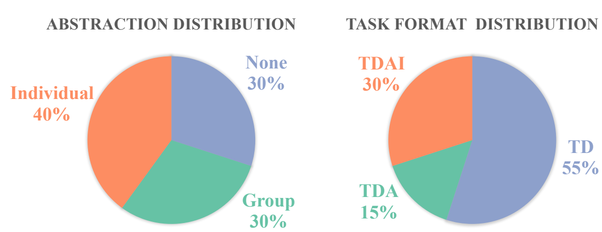


Figure 4: Abstraction Distribution corresponding to *C2: Missing or Incomplete Task Abstraction* shows the variation in task abstraction techniques. 70% of the papers had none or only group-level abstraction. Task format Distribution corresponding to *C3: Inconsistent Task Description Format* shows the variation in task description format. 55% of the papers did not present instructions and abstractions, along with the task description.

the information visualization community. We reviewed BELIV 2018 papers where replication and reproducibility were a common topic of discussion among the published papers [36, 39, 63]. However, in these papers, we did not find a thorough analysis or discussion of how task phrasing and description can play a role in experiment replication. This lack of acknowledgement by the community on the importance of task phrasing and reporting for study validity and replication, and lack of supporting guidelines on how to consistently report tasks, has caused this challenge.

Effect: Inconsistent task description in existing research papers can confuse researchers designing studies at stage (Fig. 2: **Design the Study**). Researchers may not find a precise method of how to phrase tasks and present it to their participants. This problem will further be reflected when the researchers write the paper (Fig. 2: **Paper Writing and Dissemination**). This may inhibit future researchers from understanding the evaluation design.

Evidence: For each tree visualization paper we surveyed, we reviewed the paper content and binned them into three categories based on the task description format in the paper:

- Task Description (TD): The explicit task list only communicated the tasks used in the study e.g., Fig. 3 (2).
- Tasks with Abstraction (TDA): The papers that included abstraction along with the task description e.g., Fig. 3 (3).
- Task with Abstraction and Instructions (TDAI): The papers that included both abstraction and task-wise instructions in the task description e.g., Fig. 3 (1).

It is important to note that some papers may provide an overview of the task instructions in the paper but the information is too disconnected from the task description. As the point of this challenge is to facilitate communication of the tasks, we argue that the TDAI method is most suitable for task communication because it bundles all the important aspects of the tasks together and facilitates readers to find the information more conveniently. The tagging results of this challenge in Fig. 4 (Task Format Distribution) shows that 30% (TDAI) of the papers are already using descriptive task information. But 70% (TD and TDA) miss out on the opportunity to provide their readers detailed information about the tasks in an objective and easy to parse manner.

Guidelines For Researcher: We recommend researchers to use a task description format similar to the one shown in Fig. 3 (1) because the form is most specific and leaves little or no room for speculation about how the tasks were conducted and what was the abstract intention of the researcher behind including the task.

Guidelines for Community: Consistency problems in the task description may include sub-problems such as the consistency of the task phrasing in terms of words from conventional taxonomies, e.g., "identify" and "summarize", or identify the level of detail which tasks should be displayed to study participants and other researchers. The visualization community should recognize the low-level challenges that inhibit creating a standardized task-description format across comparative studies and suggest guidelines and methods to eliminate these challenges.

Topology Query Tasks

	Action(Search)			
Action(Query)	**Lookup**	**Locate**	**Browse**	**Explore**
Identify	18	2	2	2
Compare	7		2	1
Summarize	2		4	

Attribute Query Tasks

	Action(Search)			
Action(Query)	**Lookup**	**Locate**	**Browse**	**Explore**
Identify	13	10	9	10
Compare	8		2	6
Summarize				1

Count of Tasks

1 18

Figure 5: Summary results of the task abstraction of 99 analytical tasks collected from the tree visualization papers. We break up the results for "Topology" and "Attribute" targets. The count in each cell of the tables corresponds to total number of tasks.

4.4 C4: Knowledge Gap in Task-Based Evaluations

Description: Existing comparative studies evaluate only a subset of the task design space. As discussed in Sec. 2.2, Schulz et al. argue that, similar to the visualization design space, tasks also have a design space, i.e., all possible combinations of analytical queries a user can perform with a visualization. However it has been observed in previous work that visualization evaluations often evaluate a subset of tasks [49, 55]. The issue of an evaluation limited in its coverage of the task-design space is common in tree visualization. The limited coverage of tasks in comparative studies creates an imbalance in the knowledge of the visualization technique. According to Plaisant [49], the effectiveness of a visualization should not be based on a task but rather depend on how well the visualization performs on all the relevant tasks. Through the case of tree visualization, we introduce a method to identify the task-design space by using a task-abstraction framework and present tools that will assist researchers in communicating exhaustiveness of their task-based comparative study.

Cause: A primary reason for this challenge is the lack of adoption of the concept of a design space for tasks [58]. The design space of tasks allows researchers to enumerate all the possible combinations of tasks that a user may want to perform with a visualization tool and may support easy identification of tasks that have previously been evaluated in studies.

Effect: This challenge directly affects Fig. 2: **Design the Study** and Fig. 2: **Paper Writing and Dissemination**. In Design the Study, if researchers are not adequately aware of the design space of tasks, they might fail to identify tasks that were necessary for evaluation but missed them due to an error in the task selection method. In Paper Writing and Dissemination, due to this challenge, the authors fail to communicate the evaluated task design space and open research areas. This shortcoming in communication can inhibit researchers interested in expanding task-based knowledge about visualizations through future work. This challenge may lead to future researchers conducting redundant studies.

Evidence: We collected all the tasks used in the surveyed tree visualization papers and abstracted the tasks using the Multi-Level Task Typology (MLTT) Framework [13]. We use this framework because it allows us to specify actions and targets of the analytical task, thus providing more detailed insight into both the intention (action) of the user and the item of interest (target). After abstraction, we reviewed coverage of the task design space for the target as shown in Fig. 5. We analyze "Topology" and "Attribute" separately. From these results we observe that some aspects of the tree visualization tasks are more thoroughly evaluated than others. Studies are more inclined towards *"Identify"* and *"Lookup"* tasks. While some other tasks like *"Summarize"* are not well evaluated, researchers and practitioners have little guidance on how tree visualizations support the *"Summarize"* tasks.

Guidelines for Researchers: Researchers should explicitly communicate and justify the evaluated design space. To determine the task design space, a user can use an existing task abstraction framework. For instance, in Fig. 5 we can see that a user can perform four search tasks (Lookup, Locate, Browse, Explore) and three query tasks (Identify, Compare and Summarize) as per the Multi-Level Task Typology [13]. Therefore, for the Multi-Level Task Typology, the task design space consists of 12 $(4 * 3)$ analytical tasks. Using the task design space, researchers can objectively communicate the tasks they have evaluated, as we have done in Fig. 5. As discussed in the evidence paragraph, the figure allows other researchers to see the tasks being assessed and open areas of research that the study did not evaluate.

Guidelines for Community: This challenge also presents the need for a task dataset. A task dataset, as discussed earlier in Challenge 1, is an exhaustive collection of visualization tasks. For example, a task dataset could be focused on a specific domain, a specific type of data, or a specific visual encoding. The task dataset can be a central resource for researchers conducting comparative studies to look up for tasks that have not been evaluated in the existing literature. Researchers can also add tasks they have evaluated in a comparative study to the benefit of other researchers.

5 Discussion and Future Work

In this paper, we present four task-based challenges of comparative evaluation studies identified through a hybrid method of literature review and personal experience. We found that common task-based problems we experienced in our own research have been discussed previously in different capacities by other researchers [26, 49, 55].

Furthermore, our survey and analysis of tree visualization comparative studies revealed that a large proportion of papers do not provide adequate information about the task source (C1), abstract definition of the tasks (C2), necessary information about the task procedure to support replication of the experiment (C3), and a through analysis of the open areas of the research (C4). Identification of these problems enabled us to reflect on ways researchers and visualization community can solve these problems.

Our paper provides practical guidelines to mitigate these task based challenges. Below we provide a checklist based on the **C1, C2, and C3** guidelines to assist researchers communicate tasks in their research and publications in a transparent easy to use manner:

Researcher Checklist for Publication:

Authors should,

- Explicitly mention the source of tasks, the reason for choosing the source and rationale for selecting the evaluation tasks.
- Provide task-level abstraction.
- Describe the task level procedure to ensure replication of the experiment by other researchers.

C4, as discussed Sec. 4, may not directly affect the validity of results, but the task space analysis may have an impact on the exhaustiveness of the study design. Researchers can identify early on in their study design process about tasks they might have missed. The task space analysis will also benefit the broader visualization community as it can be a source to identify potential areas of future work. Therefore, we recommend researchers should include the information of the evaluated task space as supplemental material.

Our community guidelines underscore the importance of creating a task dataset. We envision the task datasets should store an exhaustive list of analytical tasks related to visualization encoding and evaluation results demonstrating the effectiveness of encoding with a set of tasks. These datasets can be valuable resources for visualization researchers designing evaluation studies or practitioners looking for guidelines to choose the right visualization encoding, given the analytical tasks.

Given the focused scope of our survey, including only tree visualization studies, and the relatively old year of publication of some of the articles used in the study, we speculate that the evidence found regarding challenges may be preliminary and may require further investigation. Therefore, we want to evaluate the proposed problems and guidelines further. There can be multiple ways to assess the challenges. We are particularly interested in interviewing researchers who have worked with comparative studies in the past. In the expert interview, we could collect more information on the challenges other researchers have faced in comparative studies. We also could gather expert's opinions on the guidelines we proposed for the problems. The opinion may further help refine and improve the guidelines for mitigating the task-based challenges in comparative visualization studies.

Although, our proposed guidelines are derived from the lens of comparative studies, we believe the guidelines may be applicable more generally to other evaluation methods discussed in Sec. 2.3 as well as design studies. For instance, design studies with usability evaluations can also follow the proposed guidelines to ensure that they are identifying the right usability tasks and also communicating the tasks more transparently.

6 Conclusion

Appropriate task selection and transparent task communication are essential to the design of comparative studies. However, the current methods of task selection and communication in comparative studies have several shortcomings. We identified four task-based challenges that can potentially affect the validity and usability of a comparative visualization study through an analysis of existing visualization literature. Corresponding to each problem, we provide necessary details to enable visualization researchers and practitioners to recognize the cause of the challenge and have a precise understanding of how the challenge affects a comparative study. We also surveyed 20 tree visualization comparative studies to determine if they were affected by the proposed task-based problems in any capacity. Our results demonstrate that several tree visualization comparative studies lacked task source, the rationale for task selection, abstract descriptions of the domain tasks, and an underspecified task communication format. Our work proposes a checklist of guidelines to assist researchers with careful task selection and accurate task communication to ensure comparative studies in the future minimize these problems.

Acknowledgments

This work was supported by NSF CISE CRII award no. 1657466 and the Khoury College of Computer Sciences, Northeastern University.

References

[1] C. Ahlberg and B. Shneiderman. The alphaslider: a compact and rapid selector. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 365–371, 1994. doi: 10.1145/191666.191790

[2] B. Alper, N. Riche, G. Ramos, and M. Czerwinski. Design study of linesets, a novel set visualization technique. *IEEE transactions on visualization and computer graphics*, 17(12):2259–2267, 2011. doi: 10.1109/TVCG.2011.186

[3] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. Visualizing Sets and Set-typed Data: State-of-the-Art and Future Challenges. In R. Borgo, R. Maciejewski, and I. Viola, eds., *EuroVis - STARs*. The Eurographics Association, 2014. doi: 10.2312/eurovisstar.20141170

[4] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 111–117. IEEE, 2005. doi: 10.1109/INFVIS.2005.1532136

[5] R. A. Amar and J. T. Stasko. Knowledge precepts for design and evaluation of information visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):432–442, 2005. doi: 10.1109/TVCG.2005.63

[6] K. Andrews. Evaluation comes in many guises. In *AVI Workshop on BEyond time and errors (BELIV) Position Paper*, pp. 7–8, 2008.

[7] K. Andrews and J. Kasanicka. A comparative study of four hierarchy browsers using the hierarchical visualisation testing environment (hvte). In *2007 11th International Conference Information Visualization (IV '07)*, pp. 81–86, July 2007. doi: 10.1109/IV.2007.8

[8] L. Barkhuus and J. A. Rode. From mice to men-24 years of evaluation in chi. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, vol. 10, 2007.

[9] T. Barlow and P. Neville. A comparison of 2-d visualizations of hierarchies. In *IEEE Symposium on Information Visualization, 2001. INFOVIS 2001.*, pp. 131–138, Oct 2001. doi: 10.1109/INFVIS.2001.963290

[10] T. Bladh, D. A. Carr, and M. Kljun. The effect of animated transitions on user navigation in 3d tree-maps. In *Ninth International Conference on Information Visualisation (IV'05)*, pp. 297–305, July 2005. doi: 10.1109/IV.2005.122

[11] M. Borkin, K. Gajos, A. Peters, D. Mitsouras, S. Melchionna, F. Rybicki, C. Feldman, and H. Pfister. Evaluation of artery visualizations for heart disease diagnosis. *IEEE transactions on visualization and computer graphics*, 17(12):2479–2488, 2011. doi: 10.1109/TVCG.2011.192

[12] M. A. Borkin, C. S. Yeh, M. Boyd, P. Macko, K. Z. Gajos, M. Seltzer, and H. Pfister. Evaluation of filesystem provenance visualization tools. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2476–2485, Dec 2013. doi: 10.1109/TVCG.2013.155

[13] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE transactions on visualization and computer graphics*, 19(12):2376–2385, 2013. doi: 10.1109/TVCG.2013.124

[14] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner. Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, pp. 1–8, 2014. doi: 10.1145/2669557.2669559

[15] D. Byrd. A scrollbar-based visualization for document navigation. In *Proceedings of the fourth ACM conference on Digital libraries*, pp. 122–129, 1999. doi: 10.1145/313238.313283

[16] S. Carpendale. *Evaluating Information Visualizations*, pp. 19–45. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-70956-5_2

[17] M. C. Chuah and S. F. Roth. On the semantics of interactive visualizations. In *Proceedings IEEE Symposium on Information Visualization'96*, pp. 29–36. IEEE, 1996.

[18] K. A. Cook and J. J. Thomas. Illuminating the path: The research and development agenda for visual analytics. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2005.

[19] J. W. Creswell and J. D. Creswell. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.

[20] A. Crisan and M. Elliott. How to evaluate an evaluation study? comparing and contrasting practices in vis with those of other disciplines : Position paper. In *2018 IEEE Evaluation and Beyond - Methodological Approaches for Visualization (BELIV)*, pp. 28–36, Oct 2018. doi: 10.1109/BELIV.2018.8634420

[21] S. Di Bartolomeo, A. Pandey, A. Leventidis, D. Saffo, U. H. Syeda, E. Carstensdottir, M. Seif El-Nasr, M. A. Borkin, and C. Dunne. Evaluating the effect of timeline shape on visualization task performance. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3313831.3376237

[22] G. Ellis and A. Dix. An explorative analysis of user evaluation studies in information visualisation. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pp. 1–7, 2006. doi: 10.1145/1168149.1168152

[23] N. Elmqvist and J. S. Yi. Patterns for visualization evaluation. *Information Visualization*, 14(3):250–269, 2015. doi: 10.1177/1473871613513228

[24] D. Gergle and D. S. Tan. Experimental research in hci. In *Ways of Knowing in HCI*, pp. 191–227. Springer, 2014. doi: 10.1007/978-1-4939-0378-8_9

[25] D. Gotz and M. X. Zhou. Characterizing users' visual analytic activity for insight provenance. *Information Visualization*, 8(1):42–55, 2009. doi: 10.1057/ivs.2008.31

[26] G. G. Grinstein, P. E. Hoffman, and R. M. Pickett. *Benchmark Development for the Evaluation of Visualization for Data Mining*, p. 129–176. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

[27] D. M. Hilbert and D. F. Redmiles. Extracting usability information from user interface events. *ACM Computing Surveys (CSUR)*, 32(4):384–421, 2000. doi: 10.1145/371578.371593

[28] P. Irani and C. Ware. Diagramming information structures using 3d perceptual primitives. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 10(1):1–19, 2003. doi: 10.1145/606658.606659

[29] P. Isenberg, A. Bezerianos, P. Dragicevic, and J.-D. Fekete. A study on dual-scale data charts. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2469–2478, 2011. doi: 10.1109/TVCG.2011.160

[30] T. Isenberg, P. Isenberg, J. Chen, M. Sedlmair, and T. Möller. A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2818–2827, 2013. doi: 10.1109/TVCG.2013.126

[31] M. Y. Ivory and M. A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, 33(4):470–516, 2001. doi: 10.1145/503112.503114

[32] F. J.-D. and P. C. InfoVis 2003 Contest. `www.cs.umd.edu/hcil/iv03contest`, 2003.

[33] N. Kerracher and J. Kennedy. Constructing and evaluating visualisation task classifications: Process and considerations. *Computer Graphics Forum*, 36(3):47–59, 2017. doi: 10.1111/cgf.13167

[34] N. Kerracher, J. Kennedy, and K. Chalmers. A task taxonomy for temporal graph visualisation. *IEEE transactions on visualization and computer graphics*, 21(10):1160–1172, 2015. doi: 10.1109/TVCG.2015.2424889

[35] A. Kobsa. User experiments with tree visualization systems. In *IEEE Symposium on Information Visualization*, pp. 9–16, Oct 2004. doi: 10.1109/INFVIS.2004.70

[36] R. Kosara and S. Haroz. Skipping the replication crisis in visualization: Threats to study validity and how to address them : Position paper. In *2018 IEEE Evaluation and Beyond - Methodological Approaches for Visualization (BELIV)*, pp. 102–107, Oct 2018. doi: 10.1109/BELIV.2018.8634392

[37] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Seven guiding scenarios for information visualization evaluation. 2011.

[38] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task taxonomy for graph visualization. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, p. 1–5. Association for Computing Machinery, New York, NY, USA, 2006. doi: 10.1145/1168149.1168168

[39] H. Lücke-Tieke, M. Beuth, P. Schader, T. May, J. Bernard, and J. Kohlhammer. Lowering the barrier for successful replication and evaluation. In *2018 IEEE Evaluation and Beyond - Methodological Approaches for Visualization (BELIV)*, pp. 60–68, Oct 2018. doi: 10.1109/BELIV.2018.8634201

[40] J. E. McGrath. Methodology matters: Doing research in the behavioral and social sciences. In *Readings in Human–Computer Interaction*, pp. 152–169. Elsevier, 1995. doi: 10.1016/B978-0-08-051574-8.50019-4

[41] M. Meyer, T. Munzner, and H. Pfister. Mizbee: a multiscale synteny browser. *IEEE transactions on visualization and computer graphics*, 15(6):897–904, 2009. doi: 10.1109/TVCG.2009.167

[42] L. Micallef, P. Dragicevic, and J.-D. Fekete. Assessing the effect of visualizations on bayesian reasoning through crowdsourcing. *IEEE transactions on visualization and computer graphics*, 18(12):2536–2545, 2012. doi: 10.1109/TVCG.2012.199

[43] T. Munzner. A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics*, 15(6):921–928, 2009. doi: 10.1109/TVCG.2009.111

[44] T. Munzner. *Visualization analysis and design*. CRC press, 2014. doi: 10.1201/b17511

[45] S. Muramalla, R. Al Tarawneh, S. R. Humayoun, R. Moses, S. Panis, and A. Ebert. Radial vs. rectangular: Evaluating visualization layout impact on user task performance of hierarchical data. *IADIS International Journal on Computer Science & Information Systems*, 12(2), 2017.

[46] J. Nielsen. Heuristic evaluation. In *Usability inspection methods*, pp. 25–62. John Wiley & Sons, Inc., 1994.

[47] A. Pandey, H. Shukla, G. S. Young, L. Qin, A. A. Zamani, L. Hsu, R. Huang, C. Dunne, and M. A. Borkin. Cerebrovis: Designing an abstract yet spatially contextualized cerebral artery network visualization. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):938–948, Jan 2020. doi: 10.1109/TVCG.2019.2934402

[48] A. Pandey, Y. Zhang, J. A. Guerra-Gomez, A. G. Parker, and M. A. Borkin. Digital collaborator: Augmenting task abstraction in visualization design with artificial intelligence, 2020.

[49] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, p. 109–116. Association for Computing Machinery, New York, NY, USA, 2004. doi: 10.1145/989863.989880

[50] C. Plaisant, J. Grosjean, and B. B. Bederson. Spacetree: supporting exploration in large node link tree, design evolution and empirical evaluation. In *IEEE Symposium on Information Visualization, 2002. INFOVIS 2002.*, pp. 57–64, Oct 2002. doi: 10.1109/INFVIS.2002.1173148

[51] A. Rind, W. Aigner, M. Wagner, S. Miksch, and T. Lammarsch. Task cube: A three-dimensional conceptual space of user tasks in visualization design and evaluation. *Information Visualization*, 15(4):288–300, 2016. doi: 10.1177/1473871615621602

[52] R. E. Roth. An empirically-derived taxonomy of interaction primitives for interactive cartography and geovisualization. *IEEE transactions on visualization and computer graphics*, 19(12):2356–2365, 2013. doi: 10.1109/TVCG.2013.130

[53] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. *IEEE transactions on visualization and computer graphics*, 20(12):1604–1613, 2014. doi: 10.1109/TVCG.2014.2346481

[54] R. Sakai and J. Aerts. Card sorting techniques for domain characterization in problem-driven visualization research. In *EuroVis (Short Papers)*, pp. 121–125, 2015.

[55] B. Saket, A. Endert, and C. Demiralp. Task-based effectiveness of basic visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(7):2505–2512, July 2019. doi: 10.1109/TVCG.2018.2829750

[56] J. M. Santos, B. S. Santos, P. Dias, S. Silva, and C. Ferreira. Extending the h-tree layout pedigree: An evaluation. In *2013 17th International Conference on Information Visualisation*, pp. 422–427, July 2013. doi: 10.1109/IV.2013.56

[57] J. Sanyal, S. Zhang, G. Bhattacharya, P. Amburn, and R. Moorhead. A user study to compare four uncertainty visualization methods for 1d and 2d datasets. *IEEE transactions on visualization and computer graphics*, 15(6):1209–1218, 2009. doi: 10.1109/TVCG.2009.114

[58] H. Schulz, T. Nocke, M. Heitzler, and H. Schumann. A design space of visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2366–2375, Dec 2013. doi: 10.1109/TVCG.2013.120

[59] M. Schwab, S. Hao, O. Vitek, J. Tompkin, J. Huang, and M. A. Borkin. Evaluating pan and zoom timelines and sliders. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300786

[60] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, Dec 2012. doi: 10.1109/TVCG.2012.213

[61] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE symposium on visual languages*, pp. 336–343. IEEE, 1996. doi: 10.1109/VL.1996.545307

[62] J. Spool and W. Schroeder. Testing web sites: Five users is nowhere near enough. In *CHI'01 extended abstracts on Human factors in computing systems*, pp. 285–286, 2001. doi: 10.1145/634067.634236

[63] P. T. Sukumar and R. Metoyer. Towards designing unbiased replication studies in information visualization. In *2018 IEEE Evaluation and Beyond - Methodological Approaches for Visualization (BELIV)*, pp. 93–101, Oct 2018. doi: 10.1109/BELIV.2018.8634261

[64] A. G. Sutcliffe, M. Ennis, and J. Hu. Evaluating the effectiveness of visual user interfaces for information retrieval. *International Journal of Human-Computer Studies*, 53(5):741–763, 2000. doi: 10.1006/ijhc.2000.0416

[65] J. G. Trafton, S. S. Kirschenbaum, T. L. Tsui, R. T. Miyamoto, J. A. Ballas, and P. D. Raymond. Turning pictures into numbers: extracting and generating information from complex visualizations. *International Journal of Human-Computer Studies*, 53(5):827–850, 2000. doi: 10.1006/ijhc.2000.0419

[66] S. Wehrend and C. Lewis. A problem-oriented classification of visualization techniques. In *Proceedings of the First IEEE Conference on Visualization: Visualization90*, pp. 139–143. IEEE, 1990. doi: 10.1109/VISUAL.1990.146375

[67] J. S. Yi, Y. ah Kang, J. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE transactions on visualization and computer graphics*, 13(6):1224–1231, 2007. doi: 10.1109/TVCG.2007.70515

[68] C. Ziemkiewicz and R. Kosara. The shaping of information by visual metaphors. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1269–1276, Nov 2008. doi: 10.1109/TVCG.2008.171

arXiv:2003.01304v1 [cs.HC] 3 Mar 2020

Digital Collaborator: Augmenting Task Abstraction in Visualization Design with Artificial Intelligence

Aditeya Pandey*
Northeastern University
Boston, MA 02115, USA
pandey.ad@husky.neu.edu

Andrea G. Parker
Georgia Institute of Technology
Atlanta, GA 30308, USA
andrea@cc.gatech.edu

Yixuan Zhang*
Northeastern University
Boston, MA 02115, USA
zhang.yixua@northeastern.edu

Michelle A. Borkin
Northeastern University
Boston, MA 02115, USA
m.borkin@northeastern.edu

John A. Guerra-Gomez
Northeastern University
San Jose, CA 95138, USA
john.guerra@gmail.com

* Authors contributed equally

Permission to make digital or
hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Copyright held by the owner/author(s).
CHI'20,, April 25–30, 2020, Honolulu, HI, USA
ACM 978-1-4503-6819-3/20/04.
https://doi.org/10.1145/3334480.XXXXXXX

Abstract
In the task abstraction phase of the visualization design process, including in “design studies”, a practitioner maps the observed domain goals to generalizable abstract tasks using visualization theory in order to better understand and address the user's needs. We argue that this manual task abstraction process is prone to errors due to designer biases and a lack of domain background and knowledge. Under these circumstances, a collaborator can help validate and provide sanity checks to visualization practitioners during this important task abstraction stage. However, having a human collaborator is not always feasible and may be subject to the same biases and pitfalls. In this paper, we first describe the challenges associated with task abstraction. We then propose a conceptual *Digital Collaborator*—an artificial intelligence system that aims to help visualization practitioners by augmenting their ability to validate and reason about the output of task abstraction. We also discuss several practical design challenges of designing and implementing such systems.

Author Keywords
Information visualization; artificial intelligence; natural language processing; task abstraction.

CCS Concepts
•Human-centered computing → Visualization; •Computing methodologies → Artificial intelligence; *NLP;*

Task-Abstraction Frameworks

To highlight the variation in task-abstraction methodologies, we present some distinguishing characteristics of three common frameworks:

A Multi-Level Typology of Abstract Visualization Tasks [1]: A generic task abstraction framework that works well across disciplines and data-set types.

Task Taxonomy for Graph Visualization [4]: A descriptive framework for tasks in the field specific to graph visualizations. This taxonomy provides more descriptive identification of visualization goals than a generalized framework [8].

Hierarchical Task Abstraction (HTA) [11]: HTA highlights the importance of integrating context and leverages existing task abstraction frameworks in combination with a systematic analysis of user tasks, goals, and processes.

Introduction

Artificial intelligence (AI) has been used in the data information community to help improve design of visualizations [5, 10]. A visualization practitioner can get help from a variety of tools (e.g., Tableau, QlikView, SAS) [3] to select proper visual encodings. However this step must be carefully considered in the context of user goals and tasks. The visualization design process can be broadly divided into phases [6], including the design study process model [9], performing task analysis to understand domain problems, and task abstraction that aims to recast user goals from domain-specific languages to a generalized terminology for better understanding and readability [6]. Conducting task abstraction is an important but rigorous manual process that requires in-depth understanding of domain knowledge and familiarity with visualization literature [1, 4, 11]. For example, a biologist may be interested in results for tissue samples treated with LL-37 matching up with the ones without the peptide. A visualization researcher may translate this task to **compare** values between **two groups** [6]. However, to accurately perform task abstraction, a visualization practitioner must first choose the best abstraction framework and then the appropriate abstraction. A practitioner has to keep up with the ever-growing task abstraction literature [1, 4, 11] and ensure that their personal biases that might come from previous work experiences do not affect their ability to perform task abstraction.

As task abstraction is a manual and subjective phase of the visualization design process, we argue that it may be prone to human-judgment errors. For example, domain experts often serve as project collaborators to help visualization researchers and practitioners validate the task analysis and abstraction in human-centered studies. However, it is challenging to have collaborators' involvement in many situations. Furthermore, human collaborators are still prone to pitfalls like keeping pace with recent development of task-abstraction theories and priming biases. Therefore, we propose an AI-enabled *Digital Collaborator (DC)* that can serve as a feasible alternative to a human collaborator. We envision that a DC can assist visualization researchers by being up-to-date with task-abstraction frameworks to help identify the most appropriate framework for abstraction and can help validate the task analysis and abstraction process to identify judgment errors or biases. With this paper we hope to open a discussion on the advantages and challenges of building a DC for the visualization community.

Challenges of Performing Task Abstraction

We first discuss the main challenges associated with the task abstraction process.

A Wide Range of Task Abstraction Approaches: Visualization researchers have proposed various task abstraction approaches (e.g., [1, 4, 11]). On page 2 (side-bar), we discuss three common visualization task abstraction frameworks and explain how they differ. Adopting an appropriate task abstraction approach is pivotal for visualization design as it impacts the choice of visualization design and interaction idioms. However, selecting a proper task abstraction framework requires an extensive comparison of existing literature.

Interpretation of a Task Abstraction Framework: Task abstraction is a subjective evaluation of the domain experts' needs. Subjective assessments are prone to errors arising from variability in the practitioner's understanding of an abstraction framework or an innate bias such as recency bias where the task-abstraction may be influenced by recent work. Such abstraction biases can lead to a "domino" effect of errors that can only be objectively verified after prototyping [9]. Additionally, the analytic-task focused taxonomies require mastery of the terminology and definitions [4]. For example, in network abstraction, it is common to use the term *Topology* for properties related to the structure of the network. Topology is a

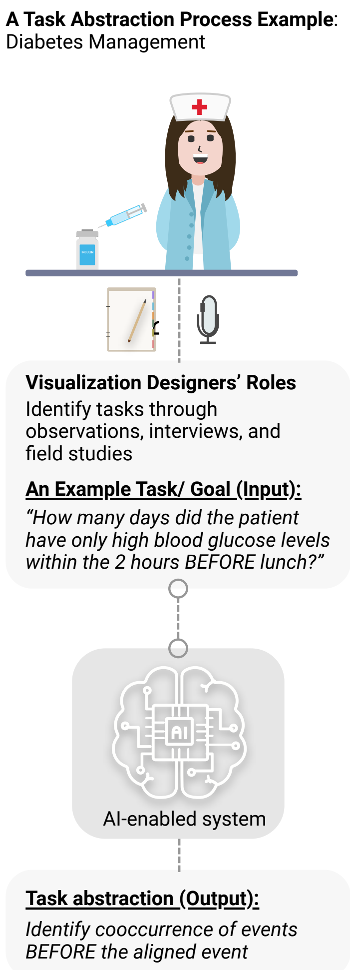


Figure 1: Example process to use AI to facilitate task abstractions in visualization research.

mathematical term, and practitioners coming from design backgrounds may be unfamiliar with its meaning.

Automate Task Abstraction using AI

We have discussed some challenges of performing task abstraction with human effort involved. Drawing inspiration from the idiom of an Intelligent Personal Assistant (IPA) [2], we propose a Digital Collaborator (DC)—a conceptual AI-enabled system to support task abstraction for visualization research. Figure 1 shows an example of how AI can be used to facilitate task abstraction.

Input and Output: Similar to the IPA system, our proposed DC will adopt a question-and-answer-based interface. The questions (*input*) will be domain goals identified by visualization practitioners through interviews and observations with domain experts. The DC should generate a translation of a domain goal to a generalized task description by applying an appropriate task-abstraction framework (*output*). To improve communication transparency, the DC should aim to provide the rationale for their output and a set of alternative translations.

AI System: We believe system goals should include identifying the right abstraction framework and recommending the appropriate analytical conversion of the tasks. Note that we do not intend to suggest replacing the human-centered approaches when conducting task analysis (e.g., field studies, interviews, and observations). Instead, we propose to leverage AI in designing systems to help ease the process of task abstraction.

Challenges of Our Proposed AI-enabled System

In order to develop an AI to help automate task abstraction we acknowledge that there will be challenges to design such an AI-enabled DC system.

Framework Characterization: Task-Abstraction frameworks are well established. However, there is little guidance on how to select the "right" framework. Therefore, the first challenge of building a DC will be to develop parameters to distinguish between these abstraction frameworks.

Training Data: For automating the task-abstraction process, we need to train machine learning models with task data and their labeled outputs. One way to acquire training data is by parsing domain goals and their abstractions from existing literature. Smart data crawling tools may facilitate the process of extracting tasks from research papers with little manual effort. However, even after deploying web-crawlers, there might be problems with data quality. For instance, there might be conflicting abstractions where similar tasks have different abstractions. To counter the problem, we can think of human-in-the-loop methodology where visualization researchers working on the project can address quality issues.

Recommendation Validation: Task abstraction involves subjective evaluation and characterization of domain problems. Practitioners may disagree with the suggested results generated by the DC. Therefore, an open question is how to instill confidence in visualization practitioners to consider the suggested results before discarding them. For example, extending the recommendation list with confidence scores may increase transparency. Future research should examine design recommendations that can boost confidence in communicating results.

Equity Issues with AI: There has been an increasing body of research on equity issues in AI research, such as biased datasets and algorithm transparency [7]. An open research question is *how can an AI-based DC system promote equity*? There is a vital need for future research to examine how such systems can be made more accessible for a wide audience. For example, how a variety of voices and experiences can be captured using such systems? How nuanced aspects of these experiences contribute to different design requirements and task abstractions, which will ultimately influence the design choices? Therefore, more research is needed to further explore these issues.

Conclusion

In this paper we propose a conceptual AI-enabled digital collaborator to assist in performing visualization task abstraction. We discuss the advantages as well as the challenges of designing such AI-enabled systems, including training data, designing for transparent communication, as well as equity issues with AI. Through this workshop paper, we want to initiate a discussion on the topic of how AI can assist task abstraction in visualization research and how to address these challenges.

Acknowledgements

This work was supported by NSF CISE CRII award no. 1657466.

REFERENCES

[1] M. Brehmer and T. Munzner. 2013. A Multi-Level Typology of Abstract Visualization Tasks. *IEEE TVCG* 19, 12 (Dec 2013), 2376–2385. DOI: http://dx.doi.org/10.1109/TVCG.2013.124

[2] Gabriela Czibula, Adriana-Mihaela Guran, Istvan Gergely Czibula, and Grigoreta Sofia Cojocar. 2009. IPA-An intelligent personal assistant agent for task performance support. In *2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing*. IEEE, 31–34.

[3] Michael Diamond and Angela Mattia. 2017. Data visualization: An exploratory study into the software tools used by businesses. *Journal of Instructional Pedagogies* 18 (2017).

[4] Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. 2006. Task Taxonomy for Graph Visualization *(BELIV âĂŹ06)*. Association for Computing Machinery, New York, NY, USA, 1âĂŞ5. DOI: http://dx.doi.org/10.1145/1168149.1168168

[5] MLUI2019. 2020. MLUI 2019: Machine Learning from User Interactions for Visualization and Analytics. Website. (02 February 2020). https://learningfromusersworkshop.github.io/.

[6] T. Munzner. 2009. A Nested Model for Visualization Design and Validation. *IEEE TVCG* 15, 6 (Nov 2009), 921–928. DOI: http://dx.doi.org/10.1109/TVCG.2009.111

[7] Osonde A Osoba and William Welser IV. 2017. *An intelligence in our image: The risks of bias and errors in artificial intelligence*. Rand Corporation.

[8] Aditeya Pandey, Harsh Shukla, Geoffrey S Young, Lei Qin, Amir A Zamani, Liangge Hsu, Raymond Huang, Cody Dunne, and Michelle A Borkin. 2020. CerebroVis: Designing an Abstract yet Spatially Contextualized Cerebral Artery Network Visualization. *IEEE TVCG* 26, 1 (Jan 2020), 938–948. DOI: http://dx.doi.org/10.1109/TVCG.2019.2934402

[9] M. Sedlmair, M. Meyer, and T. Munzner. 2012. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE TVCG* 18, 12 (Dec 2012), 2431–2440. DOI:http://dx.doi.org/10.1109/TVCG.2012.213

[10] Manasi Vartak, Silu Huang, Tarique Siddiqui, Samuel Madden, and Aditya Parameswaran. 2017. Towards visualization recommendation systems. *ACM SIGMOD Record* 45, 4 (2017), 34–39.

[11] Yixuan Zhang, Kartik Chanana, and Cody Dunne. 2019. IDMVis: Temporal Event Sequence Visualization for Type 1 Diabetes Treatment Decision Support. *IEEE TVCG* 25, 1

(Jan 2019), 512–522. DOI:
http://dx.doi.org/10.1109/TVCG.2018.2865076

Towards a Knowledge-Based Recommendation System for Genomics Visualization

Aditeya Pandey*
Northeastern University

Sehi L'Yi†
Harvard Medical School

Nils Gehlenborg‡
Harvard Medical School

Abstract

Analysis and interpretation of genomics data are the backbones of breakthroughs and discoveries in biomedical research. Visualization tools and techniques play a significant role in the workflow of genomics researchers, and they are regularly employed in the interpretation of genomics data. However, the vast majority of genomics researchers have little or no formal training in data visualization design. Therefore, they require guidance on designing effective visualizations for a given set of data and analysis tasks. In this poster, we present the methodology behind a recommender system for genomics data and our preliminary design of a knowledge-based recommendation system. The system allows genomics researchers to navigate through a selection of visualization options and identify the techniques that meet their preferences and requirements.

Index Terms: Human-centered computing—Visualization—Visualization design and evaluation methods

1 Introduction

A rapid growth in the availability of genome and epigenome data has been driving a revolution in how research in biology and medicine is conducted. Visualization plays a critical role in generating knowledge and communicating insights into genomics data in the biomedical research community. Over the years, the biomedical and visualization research communities have published hundreds of tools and visualization techniques to facilitate exploration and analysis of how genomics data are organized [4]. However, through expert interviews and surveys, we found that domain experts lack training in data visualization, thus their choice of visualization tools is often *ad hoc* and determined through trial and error. Furthermore, many domain experts are unfamiliar with the state of development in genomics visualization, and they may resort to using tools that they have been using in the past or only on the basis of the data formats that they support.

In the visualization community, knowledge-based recommendation systems incorporate design guidelines derived from existing graphical perception studies to support novice users in choosing effective visualization techniques [3,5]. However, general purpose visualization recommendation systems may not sufficiently support users in specific domains because the systems do not necessarily cover the design space of domain-specific visualizations and analytical tasks can be different.

In this poster, we present preliminary results on building a recommendation system for genomics visualization. More specifically, we identify six stages for constructing genomics visualizations and suggest how different data features and analytical tasks can affect the recommendation of different visualization designs in those stages (Sec. 3). We also suggest our approach for building a knowledge-based recommendation system for genomics visualization, which can be employed for other domains beyond genomics field (Sec. 2).

*e-mail: pandey.ad@northeastern.edu
†e-mail: sehi_lyi@hms.harvard.edu
‡e-mail: nils@hms.harvard.com

2 Recommendation Model Creation Methodology

The first step in building the recommendation system was to identify the data structures, analytical tasks, and visualization designs used in genomics analysis. The required information was gathered from the survey paper by Nusrat et al. [4], where the authors contributed a data, task, and visualization taxonomy for genomics visualization. Overall, we realized that for a domain-specific recommendation system, a taxonomy explicitly defines the input (the data and task) and output specification (visualization) of the model.

Next, we characterized the typical design workflow of a genomics visualization. As shown in Fig. 1B, to create a genomics visualization, a designer needs to make several design decisions like the choice of marks and channels to encode genomics data or how to layout the marks and channels. Our analysis found that design stages are sequential, meaning each step feeds into the next one. For instance, the choice of alignment depends on the choice of layout. Identifying typical design workflow allows us to model our recommendation system in a way that helps genomics researchers to identify components that they are familiar with.

The third step in creating a recommendation model was identification of decision rules that ultimately inform the selection of a visualization. A decision rule can be broken down into an antecedent that represents the data and task constraints of the domain and the consequent, which is the output visualization that effectively supports the data and task requirements. For instance, **antecedent:**, if the datatype is quantitative and the task is a comparison, **consequent:**, then the recommendation model should suggest the use of length channel to encode the value [1]. Decision rules that we use for our recommendation models are derived from general visualization graphical, and perception studies [1] and analysis of genomics visualization literature published at visualization conferences [2].

3 Knowledge-Based Recommendation System

Conventionally, recommendation systems consist of three main components: (1) inputs, (2) a recommendation model, and (3) recommendation outputs. In this section, we provide an overview of how do we adapt the components of recommendation systems for genomics visualization.

Data and Task Description. We consider both data features and types of analytical tasks as potential input descriptions for our recommendation model as they are pivotal to the design of genomics visualization (Fig. 1A). The data specifications for a genomics visualization inform the recommendation model about the data type of variable that needs to be visualized and other metadata information like the overall volume of the scale of the data. Such information about data can be either provided explicitly by users or collected automatically from the standard file formats used for genomics data. In addition to data-driven recommendations, our system will support task-based recommendations. Typical tasks in a genomics data analysis include comparing two feature values, identifying a correlation between two datasets, and summarizing the entire genomics data distribution [4]. Unlike data descriptions, tasks that users are intended to perform are difficult to infer, which requires task descriptions to be explicitly specified by users.

Recommendation Model. To translate the data and task descriptions into effective genomics visualizations, we propose a sequential

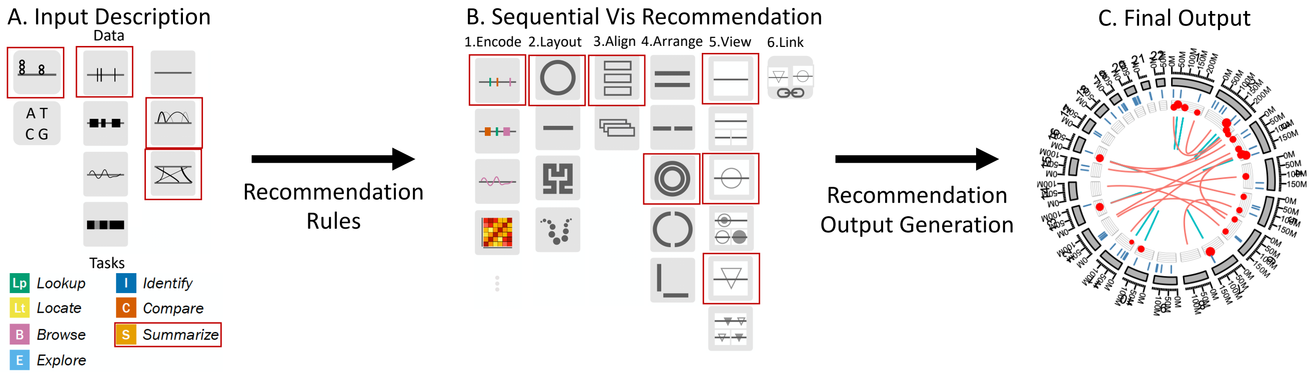


Figure 1: A schematic representation of the three stages of a visualization recommendation system for genomics data. The input visually presents the data and task specification for the system. Based on data and task requirements, the sequential recommendation model identifies suitable visual encoding at each step of the visualization grammar, and the output visualization represents the final deliverable to the user for their analysis.

knowledge-based recommendation model. We use a sequential model to mimic the common workflow of designing genomics visualizations as discussed in Sec. 2. Based on our prior work, we identified the following six main design stages for genomics visualizations (Fig. 1B).

1. Encoding Selection. The first step is the selection of appropriate visual mark or channel to encode genomics data. Based on the results of perception studies (e.g., [5]), our recommendation system suggest a visual mark and a visual channel that are most likely to be effective with the given data type and tasks.

2. Track Layout. After selecting the encoding, individual data points in a genomics dataset are arranged in a layout that preserves their sequential nature (Fig.1B-*Layout*). In practice, most genomics visualizations show the coordinate systems either linearly or circularly, but there are some alternatives, such as a space-filling layout. The correct layout selection directly affects how the visualization supports tasks such as the comparison of genomics features.

3. Alignment Method. The tracks in genomics visualizations can be parallel aligned to reduce visual clutter, or the information can be superimposed to display an overview rather than showing all individual features (Fig.1B-*Align*). The selection of alignment often depends on the type of layout users choose in *Stage 2*. Linear and circular layouts have different visual outputs for the alignment techniques.

4. Track Arrangement. Axes of the same track in the *Track Layout* can be arranged in different forms (Fig.1B-*Arrange*). The arrangement of tracks within a design is used to visualize interconnection features and compare two different sequences in comparative genomics. Therefore, the selection of track layout is governed by guidelines that inform us about the effectiveness of a track arrangement to support the connection and comparison analysis tasks [2].

5. View Configuration. A genome sequence can contain billions of nucleotides and analyzing it on different scales, or analyzing more than one region of interest at a time can be of great value. The view configuration step of the model gathers the user's high-level analytical requirements and describes the scale, foci, and views to use to visualize the genomics data (Fig.1B-*View*).

6. Linked Views. The last step in the design of a genomics visualization is the configuration of linked views. Linked views are usually defined between the views recommended at the *View Configuration* stage. This recommendation step aims to identify the interactive features to implement between the views to allow simultaneous navigation and exploration, playing a vital role in the comparative analysis of genomes.

Choosing a sequential model was a natural choice because of the incremental and dependent nature of genomics visualizations. At each stage, the recommendation model acts on a set of user-defined or implicitly determined inputs. For instance, during the *Encoding Selection*, the input consists of the feature data type and tasks. Then the decision rules at the stage determine the most appropriate visual output corresponding to the stage. The incremental model allows users to validate the visualization at the early stages of design and learn about the design practices recommended in the visualization literature.

Recommendation Output. Existing visualization tools support limited encoding, tasks, and features [4]. Therefore, we will implement a JavaScript library, a web-based framework for generating custom genomic visualizations. It will enable the recommendation model to easily create and deploy custom interactive multi-scale visualizations as a web-based tool. A customized recommended visualization output is shown in Fig. 1C.

4 Conclusion and Future Work

This poster presents a recommendation system design that aims to support genomics researchers with limited visualization experience to generate valid and meaningful genomics visualizations. Our preliminary model design methodology highlights the importance of surveying data and task requirements to create a domain-specific recommendation system. Furthermore, the proposed recommendation system uses a sequential recommendation model, that allows researchers to develop the visualization incrementally and adds transparency to the visualization recommendation process. We are currently designing the user interface of the recommendation tool that will be used for genomics researchers. Additionally, we are working on broader research questions such as the method of task input for a visualization recommendation. We also aim to validate the proposed model with more domain experts. Overall, we believe our proposed method will enable the development of a reliable and suitable visualization recommendation tool for genomics researchers.

References

[1] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*.

[2] M. Meyer, T. Munzner, and H. Pfister. Mizbee: A multiscale synteny browser. *IEEE TVCG*. doi: 10.1109/TVCG.2009.167

[3] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE TVCG*. doi: 10.1109/TVCG.2018.2865240

[4] S. Nusrat, T. Harbig, and N. Gehlenborg. Tasks, techniques, and tools for genomic data visualization. *CGF*. doi: 10.1111/cgf.13727

[5] B. Saket, D. Moritz, H. Lin, V. Dibia, C. Demiralp, and J. Heer. Beyond heuristics: Learning visualization design. *arXiv:1807.06641*.

Segmentrix: A Network Visualization Tool to Develop and Monitor Micro-Segmentation Strategies

Aditeya Pandey* Northeastern University
Larry Chan† Illumio Inc.
Raven Gao‡ Illumio Inc.
Joy Scott§ Illumio Inc.
Brian Staats¶ Illumio Inc.

ABSTRACT

Micro-Segmentation enables organizations to logically divide the datacenter into distinct security segments down to the individual workload level, and then define security controls for each unique segment. Micro-segmentation allows flexible organization of network assets into meaningful groups. For example, workloads in a network can be divided into production and development environments, and policy can control communication between the environments. Network visualization plays a critical role in the development and maintenance of segmentation. In an unsegmented network, a network visualization of workload communication can help domain users assess dependencies and create segmentation policies. Whereas, in segmented networks, the visualization of traffic between individual workloads and segmented groups can be essential for monitoring security compliance. We present a systematic overview of micro-segmentation visualization goals and use the goals to develop Segmentrix a novel tool which aids organizations to segment and monitor their networks.

Index Terms: Human-centered computing—Visualization—Visualization techniques—Adjacency-Matrix; Human-centered computing—Visualization—Visualization design

1 INTRODUCTION

Micro-segmentation is an emerging security practice of applying security controls to the datacenter and cloud assets that have an explicit business purpose for communicating with each other. Micro-segmentation is built on the principle of flexibility, which makes it different from traditional network segmentation and firewall implementation. Flexibility expresses security policies in abstract but meaningful concepts (such as web, application, and database tiers) rather than in terms of network constructs (such as IP addresses, subnets, and VLANs).

Visualization plays a crucial role in the development and analysis of segmentation strategies [1–3]. Previously, adjacency matrices have been used to visualize network segmentation with constructs like IP addresses [2, 3]. Kim et al. [2] justify the use of matrix based on three parameters: scalability with data, readability of nodes, and visibility of links. These factors also extend to the micro-segmentation goals. More specifically, readability and visibility of the network are essential for developing segmentation strategies. Readability ensures the users have context when analyzing nodes, like information about the critical nodes, and visibility of links reduces the chances of missing vulnerable connections [4] while writing security policies.

We contribute a systematic overview of micro-segmentation visualization design goals. And use the goals to develop Segmentrix: a novel network visualization that supports development and monitoring of micro-segmentation strategies.

*e-mail: pandey.ad@husky.neu.edu
†e-mail: larry.chan@illumio.com
‡e-mail: raven.gao@illumio.com
§e-mail: joy.scott@illumio.com
¶e-mail: brian.staats@illumio.com

2 DOMAIN GOALS

We discuss micro-segmentation domain goals from the context of development of strategies and monitoring of network post-segmentation.

Develop: To develop micro-segmentation strategies, users need visibility of their network and an interface to write security policy to establish communication protocol. Therefore, we identify two tasks a visualization tool should support for development of micro-segmenation strategies: **Goal 1**: Visualize dependencies and traffic flow between workloads. **Goal 2**: Segement(divide) the network into groups and write security policies.

Monitor: Segmentation is a manual task, to support post-segmentation validation and monitor strategies, we recognize that a visualization tool should support the following tasks. **Goal 1**: Visualize connection between segments. **Goal 2**: Drill down on segments to analyze them in isolation. **Goal 3** Update Policies.

3 DATA AND VISUAL ENCODING

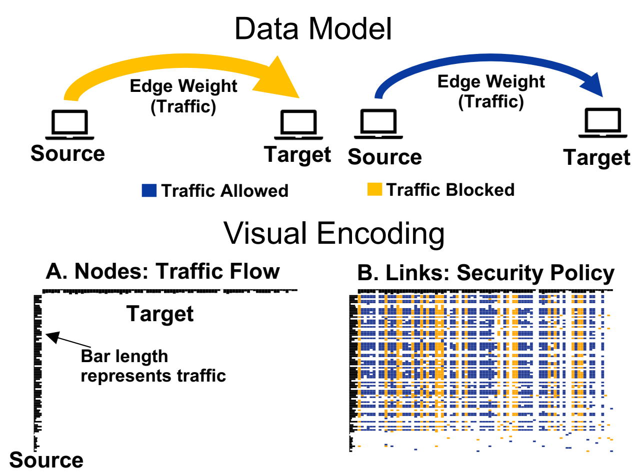


Figure 1: Micro-segmentation uses network data, where nodes represent workloads and a connection has two attributes traffic and policy. For the visualization, source node and target nodes are placed separately on the vertical and horizontal axes. The bar length encodes traffic between the source and target nodes. Further, the security policy is represented by the color-coded cell in the matrix.

Data Model: Datacenter network is a directed network. In the directed network, a workload(source node) requests a client(a target) for information. The client responds based on the established security policy. In Fig. 1: Data Model, a link in the network encodes total traffic flow and the state of the policy (Fig. 1: Data Model).

Visual Encoding: In Fig. 1: Visual Encoding, we explain the mapping of network data to an adjacency matrix. Nodes are explicitly represented on the vertical(source) and horizontal(target) axes, and each node represents the total number of requests as the bar length. Matrix cell encodes the policy decision set by the security analyst.

Data: In the poster abstract, we use randomly generated data.

4 SEGMENTRIX

Segmentrix is a network visualization tool that supports micro-segmentation task. The tool supports exploration of the network, segmentation of the workloads into meaningful groups, defining security polices and analysis of segments using filtering and interaction.

Fig. 2 shows a typical workflow adopted by a Security Analyst in developing segments and writing security policy. The task involves three main activities, exploring the network, grouping of workloads and writing security policy. Segmentrix uses an adjacency matrix to show the network. Interaction in Segmentrix allows exploration of links and traffic. Users can order nodes based on attributes like traffic. Segmentrix supports sorting of nodes by traffic, moving the most connected nodes to the top and left corner of the interface (Fig. 2 B). After, analyzing the network users can segment or divide the nodes into meaningful groups (Fig. 2 C). For example, all the workloads which store credit card information can be placed in one group and isolated with security policies from the rest of the network. For grouping and policy writing, Segmentrix allows users to define labels for nodes and policies for links. Segmentation and policy implementation can also be done outside the tool with advanced segmentation softwares.

Figure 2: Develop Segmentation Strategies: Sub-Fig. A shows an unsegemented network, in Sub-Fig. B nodes have been sorted by the traffic volume for each node. In Sub-Fig. C, we demonstrate logical segments created by the security analysts. And in Sub-Fig. D we show that security analyst secures communication between workloads based on network segments.

Post segmentation, analysts can use Segmentrix dashboard to validate and update their security policies. Sometimes analyst may want to explore the topology of the network. To ease the visualization of topology, we display a node-link visualization of a selected segment in the adjacency matrix. In Fig. 3, the highlighted source workloads, and all the connected target workloads are displayed as a node-link visualization in the linked widget.

A crucial task in monitoring of network is the ability to drill down on data and analyze segments of interest. Segmentrix allows users to filter the data by grouping label, for example in Fig. 4, the user has filtered all the source workloads which were grouped as 'Production'. The filtered view can useful for anomaly detection. As the analyst is dealing with a smaller focused group of workloads, they can look for patterns that may have been missed in the overview mode of Segmentrix.

Figure 3: Segmentrix supports interaction to analyze segments. In this figure, the user has selected a segment to analyze. The selection highlights the source segment and all the connected target segments. For context, we also display the node-link visualization of the selected segments.

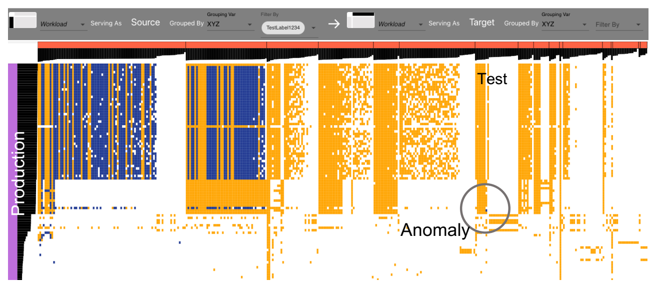


Figure 4: Segmentrix allows filtering data to analyze segments in isolation. The navigation bar at the top shows the label XYZ as the grouping variable. Further, the user has filtered out TestLabel1234 from the dataset to analyze in isolation. After filtering, we notice an anomaly, where all but one connection is allowed between segments TestLabel1234 and Test.

5 CONCLUSION

We present an overview of micro-segmentation visualization goals and the corresponding network definition of the datacenter. We developed Segmentrix, a novel adjacency matrix-based tool for developing and monitoring micro-segmentation strategies. This representation is scalable, readable, and provides visibility into the datacenter network of large organizations. The ability to visualize all the network level dependencies in one view makes it an essential tool for developing segments in the datacenter. To support monitoring of the network, we provide a linked interactive dashboard with functionalities like data filtering. As the size of datacenter grows and networks become more flexible, the need for micro-segmentation will rise and thus we expect to see more work in the domain in the near future.

REFERENCES

[1] J. Friedman. *The Definitive Guide to Micro-Segmentation*. 2014.
[2] H. Kim, S. Ko, D. S. Kim, and H. K. Kim. Firewall ruleset visualization analysis tool based on segmentation. In *2017 IEEE Symposium on Visualization for Cyber Security (VizSec)*, 2017. doi: 10.1109/VIZSEC.2017.8062196
[3] H. Koike, K. Ohno, and K. Koizumi. Visualizing cyber attacks using ip matrix. In *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05).*, 2005. doi: 10.1109/VIZSEC.2005.1532070
[4] Y. Liu and H. Man. Network vulnerability assessment using bayesian networks. In *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005*. International Society for Optics and Photonics, 2005. doi: 10.1117/12.604240