

# 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,j.guerragomez, m.borkin}@northeastern.edu

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

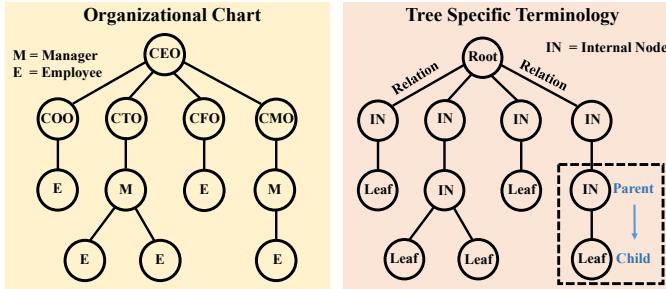


Fig. 1. Left: An organization chart as a tree visualization that shows the structure of an organization and the relationships and relative ranks of its parts and positions/jobs. Right: In the graph theory literature, the nodes and links of trees are identified by a special set of terms based on their 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 213 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 advocacy 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<sup>1</sup>. 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 the 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 dendograms. 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.

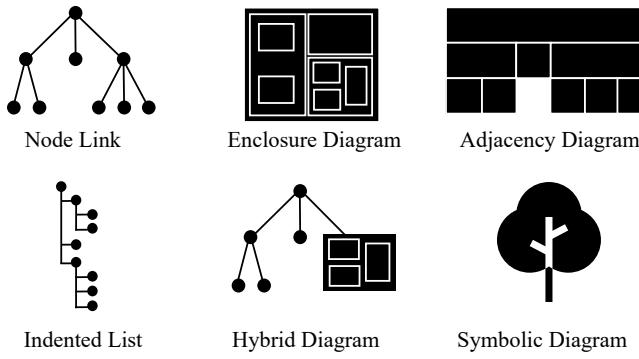


Fig. 2. Common tree visualizations grouped by parent-child encoding. More information regarding each encoding is available in Sec. 2.3. This figure is partially adapted from the work of Zhao et al. [30].

**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. In our survey, we use the node-link diagram as a category for representing all the above discussed visualizations. We do not explicitly include the variations within node-link diagrams such as 3D or hyperbolic representation.

**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 static 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 213 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 subset 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

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
		$\Sigma 1061$

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.

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 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 “single-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

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 & McKen- zie [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

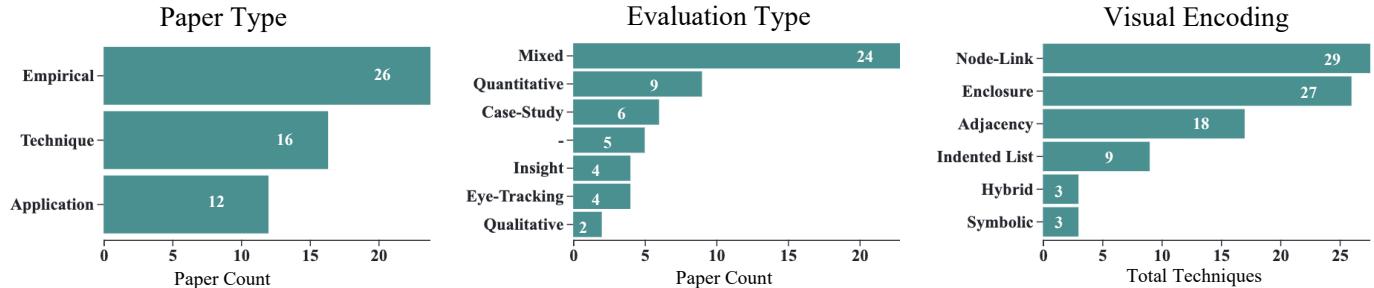


Fig. 3. In this figure we present an analysis of surveyed papers dataset (Sec. 3.3). The left-most Paper Type plot shows task data is spread across a wide-range of paper types. The Evaluation Type plot shows the specific types of evaluations included in the surveyed papers. In the evaluation type plot,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

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 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 onward, 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. We also found one “Theory/Model” [12] paper, the paper had theoretical information about tree visualization tasks, but did not contain specific information on how the tasks will be applied or used. 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,

## Tree-specific Extension to the Multi-Level Task Typology Framework

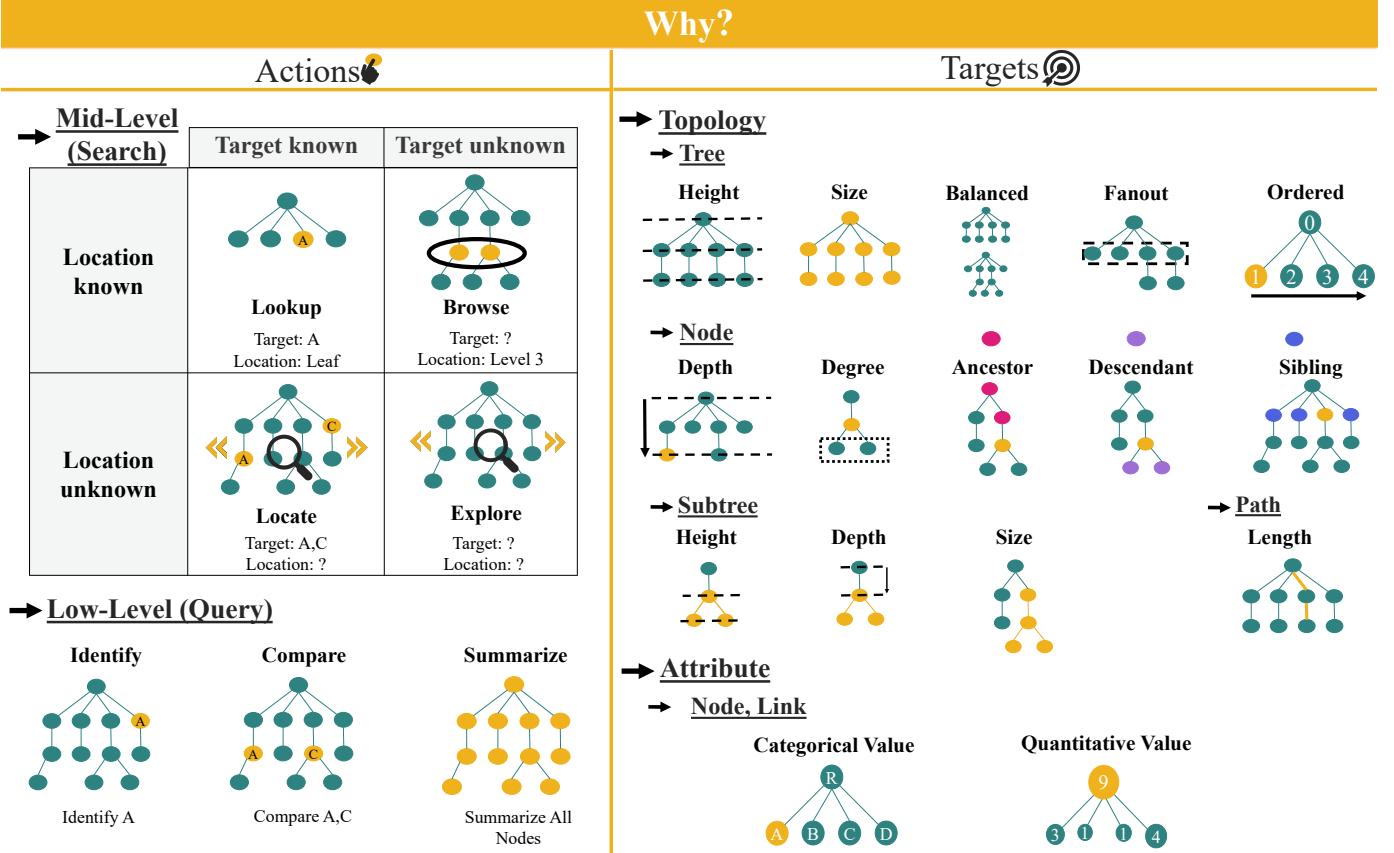


Fig. 4. Summary of tree visualization tasks broken down by *Actions* and *Targets*. The Actions use the Multi-Level Task Typology (MLTT) [1], [2] terminology to identify the types of actions users can perform in tree visualization tasks. The Targets include a novel *Nested-extension* of the existing MLTT target characterization that adds specificity for tree visualizations. Due to the skewness in high-level action classification data, majority of the tasks were of the type “consume” and “discover”, we do not include all the available high-level classification options in this figure.

“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 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:

- How many documents are available on the topic of The Governments of Thüringen? [86]
- How many detected pesticides in leafy vegetables in Tianjin city? [69]

both translate to the same abstraction: “Identify the degree of a node.”. 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 identified one “produce” task. In the task, users were allowed to annotate

regions of a tree visualization [57]. Remaining 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 because high-level tasks are often context-dependent and generally papers include insufficient discussion about the visualization creator’s goals (high-level actions) 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 classification of the type of search also depends on an individual’s spatial memory and domain expertise. However, in this survey, we are not making any assumptions about users’ prior knowledge. As a result, we only depend on the task descriptions that we found in the surveyed articles for the abstraction of tasks. 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

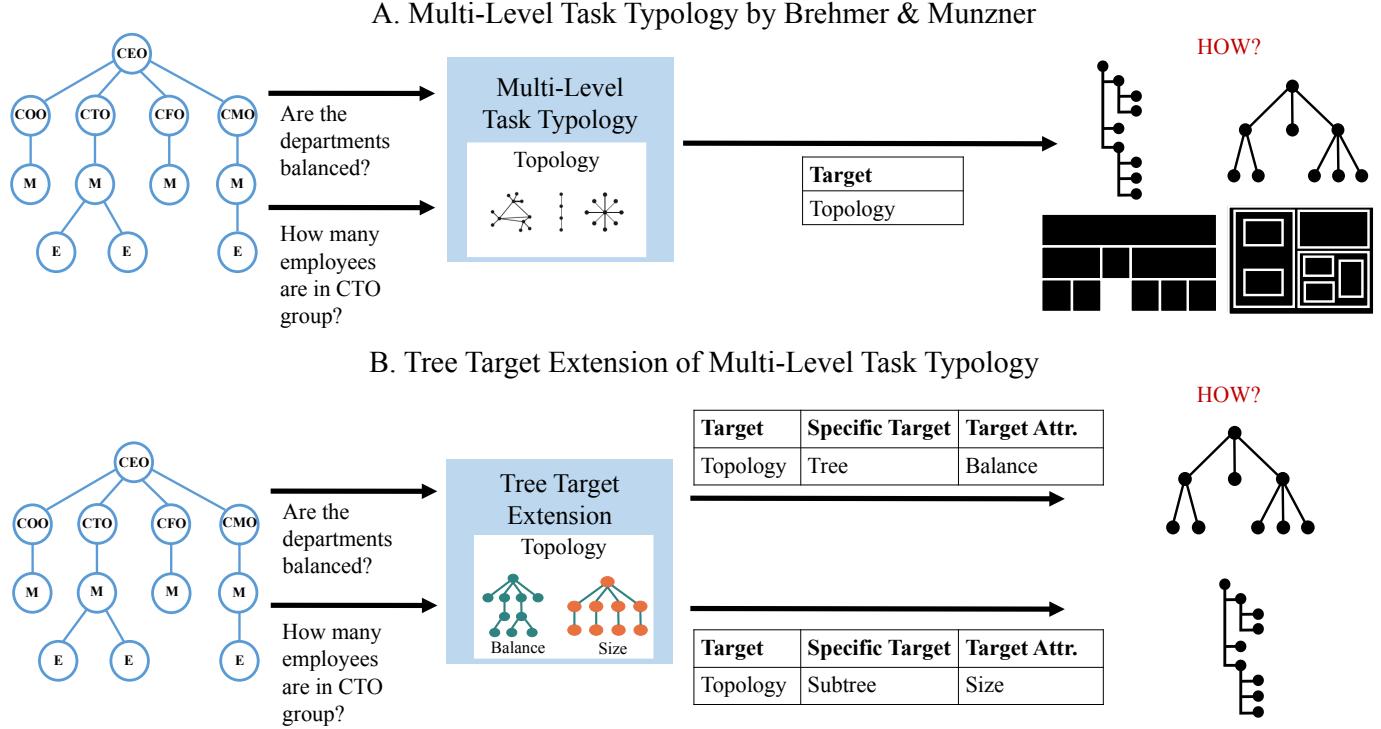


Fig. 5. The figure (A) shows how Multi-Level Task Typology abstracts two tasks which are related to different topological attributes(balance and size) to a blanket topology category. High-level target of topology limits our understanding of target of interest. Consequently, the abstract definition does 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.

to pick. She could in theory choose a list diagram, an adjacency 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:

**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 these 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 (48.35%) 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.

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



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 specific target and target attributes. In this table we also list example tree visualization tasks corresponding to each target attribute. We have generated these task examples independent of domain jargon to facilitate easy comprehension of the analytical goals. 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.

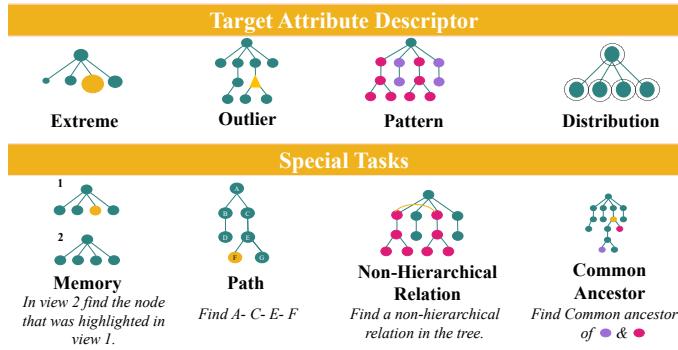


Fig. 6. Examples of *Target Attribute Descriptors* (Sec. 4.4) and *Special Tasks* (Sec. 4.5). For each special task, we also present a sample task description.

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*)

**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 213 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. To abstract sequenced tasks, MLTT [2] proposes a method of chaining the tasks as a sequence of interdependent steps and using the output from each step as an input to the next step. For instance, in the above example the first the user should look at the entire tree (input) and successfully locate the subtrees “Flutes” and “Guitars” (output). The subtrees output from the first step then becomes input to the second step and in this step the user compares height of the two input subtrees. The sequential analysis of interdependent tasks makes it easy to understand all the underlying goals of a task, and can help visualization creators create user interface that support such a sequential analysis. For the abstraction of such tasks, we initially broke down the tasks into granular unit tasks as suggested by MLTT typology. 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 formatted tasks. Both tasks: “How many documents are available on the topic of The Governments of Thüringen?” [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

### Targets Tally

Target	Specific Target	Target Attribute	# of Tasks
Topology (Tasks = 110)	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
Attribute (Tasks = 103)	Subtree (Tasks = 31)	Size	20
		-	6
		Height	4
		Depth	1
	Path (Tasks = 5)	-	4
		Length	1
		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.

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 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 crowdsourced 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.

**Underrepresented tree visualization tasks in the survey:** In the tally of mid-and low-level actions (results included in Supplemental Material), we noticed a disproportionately low number of “summarize” (4/213) tasks. We speculate that the inclusion of evaluation papers (26/54) resulted in a low number of summarize tasks. Generally, evaluation papers include tasks that are specific enough to measure completion time and accuracy. Summarize tasks may not be directly tied to a particular insight and can be hard to validate. We believe this leads to lower adoption of summarize tasks in evaluation papers. We also believe that the lack of summarize tasks affects the types of targets evaluated for tree visualizations. For instance, order tasks are generally summarize tasks where the user has to understand if there is an underlying pattern in the position of the nodes. We only found one order task in our survey, and we believe the lack of inclusion of summarize tasks affects inclusion of order tasks, too. High-level goals of a visualization include “Produce” or “Consume” goals [1], [2]. In our survey, the data for high-level actions was skewed against consume tasks, and we only found 1 produce task. Within produce, we also noticed that there were no tasks of the nature derive and record. “Enjoy” tasks are also underrepresented in our survey. 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. These underrepresented task actions, including “produce”, “summarize” and “enjoyment”, provide opportunities for future tree visualization research.

**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 213 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, van 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]. 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 213 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. This manuscript benefited greatly from the comments of anonymous reviewers. We are 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. Alsalakh, 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.
- [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)*. 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.

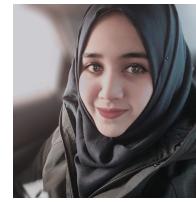
- [39] M. Harrigan and P. Healy, "Efficiently drawing a significant spanning tree of a directed graph," in *2007 6th International Asia-Pacific Symposium on Visualization*, Feb 2007, pp. 53–59.
- [40] K. Kucher and A. Kerren, "Text visualization techniques: Taxonomy, visual survey, and community insights," 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.
- [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 dendograms," 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: Treiversity2 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 *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. Hößfeld, 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.

**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.

