

Revisiting the Design Patterns of Composite Visualizations

Dazhen Deng¹, Weiwei Cui¹, Xiyu Meng, Mengye Xu, Yu Liao, Haidong Zhang, and Yingcai Wu²

Abstract—Composite visualization is a popular design strategy that represents complex datasets by integrating multiple visualizations in a meaningful and aesthetic layout, such as juxtaposition, overlay, and nesting. With this strategy, numerous novel designs have been proposed in visualization publications to accomplish various visual analytic tasks. However, there is a lack of understanding of design patterns of composite visualization, thus failing to provide holistic design space and concrete examples for practical use. In this article, we opted to revisit the composite visualizations in IEEE VIS publications and answered what and how visualizations of different types are composed together. To achieve this, we first constructed a corpus of composite visualizations from the publications and analyzed common practices, such as the pattern distributions and co-occurrence of visualization types. From the analysis, we obtained insights into different design patterns on the utilities and their potential pros and cons. Furthermore, we discussed usage scenarios of our taxonomy and corpus and how future research on visualization composition can be conducted on the basis of this study.

Index Terms—Datasets, visual analytics, visualization specification, visualization design

1 INTRODUCTION

DATA visualizations aim to visually represent data attributes to efficiently achieve the goals of analysis or storytelling [1]. For a long time, common visualizations (e.g., bar charts, line charts, and scatter plots) have been well-accepted by the public and widely adopted in business, education, and scientific research. Because of the advancement in technology, complex data (e.g., large-scale, heterogeneous, hierarchical, and spatio-temporal) have become more and more available, and visualizations have also been evolving along with the complexity of analysis tasks, leading to the bloom of the visualization research community. To address challenging analysis tasks, novel visual representations have been proposed from time to time, but the majority of research in this field still focuses on existing visualizations. One common practice of leveraging existing visualizations for complex tasks is to compose different visual representations to exploit their advantages and make up for their disadvantages [2], [3], [4], [5]. In this work, we use the term *composite visualizations* to describe these visualizations. Basically, they are a type of visualization that combines multiple

visualizations in a meaningful and aesthetic layout [2], such as juxtaposition, overlay, and nesting, to fulfill the need for specific data structures, analysis tasks, and usage scenarios.

Composite visualizations cover a large variety of design patterns. For example, a common design pattern is juxtaposing multiple charts side-by-side, which is flexible in layout and easy to understand, even for visualization novices. To improve visual coherence, juxtaposed charts can be arranged in specific patterns, such as sharing an axis or repeatedly listing the same types of visualizations (e.g., Fig. 5). In this way, a complex dataset can be visualized with multiple simple charts exhibiting different aspects of the data. Because of the ease of implementation and understanding, the juxtaposition is widely used in visual analytics systems [6], fact sheets [7], visual data stories [8], etc. Apart from juxtaposition, multiple visualizations are often compacted into a single view by overlaying (e.g., Figs. 6 and 7) or nesting (e.g., Fig. 8). By correlating the spatial and semantic relationships between graphical elements, such an integrated visualization is mainly tailored to reveal a specific type of pattern of the back-end data. However, designing a successful composite visualization is not an easy task. It requires not only an extensive knowledge base of visualization charts but also sufficient design skills to coherently present graphical elements for analysis tasks.

On the other hand, there is a growing collection of visualization publications containing well-designed composite visualizations, which serve as a resource to reuse and inspire new research. To design a new visualization, enumerating different combinations of visualizations will take a lot of time to edit, but may not necessarily result in a design with promising visual effects. It will be more cost-friendly to leverage concrete design examples for further adaptation to a new task, instead of starting from scratch. From the perspective of research, a holistic design space of the composite visualizations might lead to new research topics (e.g., which

- Dazhen Deng, Xiyu Meng, Mengye Xu, Yu Liao, and Yingcai Wu are with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, Zhejiang 310000, China. E-mail: {dengdazhen, mengxiyu, mengyexu, yuliao, ycwu}@zju.edu.cn.
- Weiwei Cui and Haidong Zhang are with Microsoft Research Asia, Beijing 100000, China. E-mail: {weiwecui, haizhang}@microsoft.com.

Manuscript received 5 May 2022; revised 26 September 2022; accepted 27 September 2022. Date of publication 11 October 2022; date of current version 10 November 2023.

This work was supported in part by NSFC under Grant 62072400, in part by the Collaborative Innovation Center of Artificial Intelligence by MOE and Zhejiang Provincial Government (ZJU), and in part by Zhejiang Lab under Grant 2021KE0AC02.

(Co-corresponding authors: Yingcai Wu and Weiwei Cui.)

Recommended for acceptance by R. Chang.

Digital Object Identifier no. 10.1109/TVCG.2022.3213565

are the most frequently used design patterns, and what are the reasons behind them?) and novel designs (e.g., is it possible to create efficient designs for specific tasks from the rarely occurring visualization combinations?).

In this work, we revisit the composite visualizations in VIS publications and try to understand their design practices from two perspectives. First, from the perspective of visual components, *what visualizations can be composed together as a composite visualization?* Prior studies have explored composite visualizations in specific contexts (e.g., visual comparison [5], [9]), data (e.g., dynamic network [10], multi-variate graph [11]), or layout (e.g., juxtaposed views [6]). We opt to answer this question from a broadened scope of visualizations beyond specific tasks and data. Removing these restrictions, we can focus on visual designs and provide valuable exemplars for visualization development. Second, from the perspective of spatial relations, *how can different visualizations be composed together?* Prior studies [2], [10], [12] have summarized different design patterns for composite visualizations. On the basis of these insights, we opted to revisit the patterns and conduct a quantitative analysis on how frequently different patterns are used [2]. Such a demographic analysis will be helpful for spotting design trends, proposing design suggestions, and discovering the potential of under-explored design patterns. Especially for researchers, a comprehensive survey can provide an overview of the community and inspire further research. For example, a widely used design pattern might request further research of empirical studies to validate its efficiency.

In this work, we first construct a composite visualization corpus from IEEE VIS publications and decompose their designs into basic visualizations. The decomposition enables us to identify composite visualizations and answer the first question. Next, based on the decomposed visual designs, we formulated a two-level taxonomy of composition patterns to answer the second question. In the taxonomy, the design patterns are identified according to the spatial relations and the semantic information conveyed. To obtain an overview of the corpus, we revisited the visual designs based on the taxonomy and conducted statistical analysis on different design patterns. Finally, we construct an exploration system for the composite visualization corpus. The system supports retrieving visualizations by type, composition pattern, and meta information. The corpus, design pattern taxonomy and the exploration system can be viewed online: <https://composite-visualizations.github.io/>. The contributions of this work include:

- A taxonomy of composition pattern and a corpus with 1,748 composite visualization examples from IEEE VIS publications.
- An in-depth analysis of the statistics, utilities, advantages, and disadvantages of different composition patterns.
- Discussions of usage scenarios of the taxonomy and future research opportunities.

2 RELATED WORK

This section introduces related studies about visualization configuration, design space of composite visualizations, and figure analysis to publications.

2.1 Visualization Configuration

Visualization configuration is a fundamental problem for visualization design and generation. Previous studies on visualization configuration have extensively studied the composition of graphical elements for visualization rendering. For example, Blackwell and Engelhardt [13] termed the “composition” as the structure of graphical primitives (e.g., line and point). Engelhardt [14] further introduced a framework of syntax that recursively formulates a visualization to be a composite graphic object. Based on the syntax, Engelhardt and Richard [15] investigated the “DNA” of visualization and proposed a grammar named VisDNA. The study indicated various relationships between graphical primitives, such as grouping, nesting, and connecting, and scaling patterns of primitives, such as repeating. Sedig and Parsons [16] proposed a language that characterizes visual design patterns and the manners to fuse the patterns, such as self-similar nesting, layering, and juxtaposing. Our work is similar to these ones in terms of leveraging the concept of “composition”, but we focus on an entirely different granularity. Specifically, the composition patterns in these previous studies mainly focus on graphical primitives, such as line, point, and circle, which are more fundamental than the ones studied in this work. For example, a bubble treemap is composed of self-nesting circles. In this work, the building blocks are high-level visualizations, which are the configurations of multiple graphical primitives. For example, a clustering heatmap is composed by stacking a heatmap and a clustering tree.

Visualization programming languages, such as ggplot (grammar of graphics [17]) and Vega-Lite [18], also investigated how graphical primitives are composed during chart rendering. In terms of composite visualizations, Vega-Lite supports fusing multiple charts by specifying the key of “concat,” “layer,” and “facet” in a declarative manner. Nevertheless, Vega-Lite mainly focuses on the composition of charts with the Cartesian coordinate system. More complex compositions such as nesting are not considered. This study opts to revisit the design patterns of composite visualizations and provide insights into the design of grammar for more powerful visualization programming languages.

2.2 Design Space of Composite Visualizations

Javed and Elmqvist [2] proposed the term *composite visualization view* as a theoretical model. In their model, a composite visualization is described by its visual components, composition pattern, and data relationship. Among them, the composition pattern, such as juxtaposition, integration, superimposition, overloading, and nesting, is used to describe how two visual components are spatially combined together (Fig. 1). Inspired by their work, we use this model as a starting point and aim to revisit the design patterns of composite visualizations in IEEE VIS publications. On top of that, our work can better identify composition patterns, obtain new insights about composite visualizations, and facilitate new usage scenarios. First, we have proposed a more refined taxonomy for revisiting the large corpus of visual designs in IEEE VIS. For example, we differentiate the overloaded views (e.g., Fig. 1 B) and superimposed views (e.g., Figs. 1 C and 1 D) considering the use of

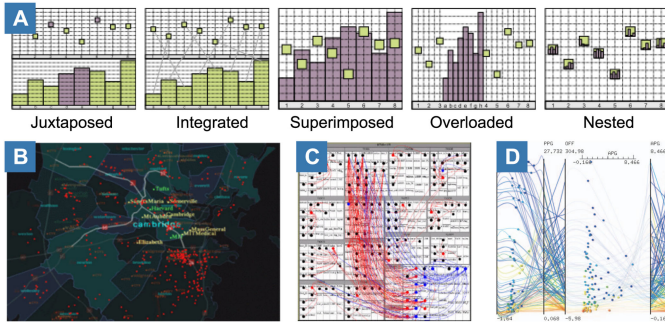


Fig. 1. Taxonomy of composition patterns (A) proposed by Javed and Elmqvist [2] and examples [19], [20], [21] of superimposed views (B) and overloaded views (C & D) provided in their paper.

coordinate systems in our taxonomy. Second, we obtained new insights into design patterns with quantitative analysis. For example, a correlation analysis on visualization types within individual composition patterns presents a general usage preference of visualization combinations in the visualization community. Third, the taxonomy and corpus can facilitate new usage scenarios, such as helping visualization designers and researchers in improving the efficiency of survey and design with an exploration system.

Other studies related to visualization composition may target specific data [10], [11], [12] or tasks [5], [22]. Researchers have explored how to combine different visualizations to represent data with specific structures or types. For example, for dynamic graphs, Beck et al. [10] studied how node-link diagrams and timelines are juxtaposed, superimposed, and integrated together to encode the temporal information. For geospatial networks, Schöttler et al. [12] analyzed the combination of node-link diagram and map based on the taxonomy of Javed and Elmqvist's [2]. Nobre et al. [11] studied the juxtaposed, integrated, and overloaded patterns of matrix and node-link diagrams in multivariate networks. Another group of studies focuses on how different compositions of visualizations affect the efficiency of a specific task. For example, Gleicher et al. [22] proposed three composition patterns of visualizations for visual comparison, including juxtaposition, superimposition, and explicit encoding. L'Yi et al. [5] further reviewed and summarized visual comparisons under these patterns, and presented several practical design guidelines.

Compared to these studies, we removed the constraints of specific data or tasks, and analyzed the design patterns of composite visualizations with an extended scope. Our study presents a corpus including diverse visualization types and layouts and an overview of the state-of-the-art composite visualizations.

2.3 Figure Analysis of Visualization Publications

In addition to survey papers, researchers also analyzed visualization publication figures. Li et al. [23] conducted a memorability study with SciVis figures. Zeng et al. [24] contributed VISTory, a technique for exploring figures in VIS publications. Chen et al. [25] adopted object detection models to extract the figures and tables in IEEE VIS publications and proposed VIS30K. These studies mainly focus on perception tasks and analytical techniques for figures, instead of the visual designs inside.

For visual designs, some studies explore how different visualizations are distributed in the figures. Deng et al. [26] collected figures from VIS publications and annotated the types and positions of visualizations. However, they only considered the co-occurrence of visualizations in the figures and failed to answer how different visualizations are composed, which is more useful for designers and researchers. Chen et al. [6] explored the composition and configuration patterns of multiple-view visualizations (MV) consisting of juxtaposed views. They discovered “many novel designs with compound view types,” which indicates more complex but under-explored design patterns other than juxtaposition. Therefore, in this work, we moved a step further and studied the composite visualizations within single views, including the types of visual components and design patterns.

3 TERMINOLOGY

A composite visualization is defined by basic visualizations and composition patterns.

- *Basic visualizations* are the components (or building blocks) of composite visualizations, referring to different types of visual representations, such as bar chart, parallel coordinate plot, and map. It is noted that a basic visualization can be broken into smaller elements, such as marks, axes, and legends, but in composite visualizations the smallest building block is basic visualization.

A series of studies attempt to classify visualizations [28], [29], [30], [31], [32]. For example, Harris [30] presented an exhaustive categorization of visualization types and indexed them by alphabetical order. Meirelles et al. [32] categorized visualizations by data structures. Heer et al. [31] classified the charts by their data structures and tasks. We choose Borkin's taxonomy [33] as the visualization type taxonomy because it covers most of the aforementioned taxonomies and contains additional up-to-date visualization types. As demonstrated by a previous study [26], this taxonomy can serve as a useful vocabulary for researchers to classify the visualizations in visualization research publications. Borkin's taxonomy classifies visualizations according to data structures, visual encodings, and tasks. The taxonomy has two classification levels (12 first-level categories, each with several second-level sub-types).

We identified three issues when using Borkin's taxonomy for the goal of this work. First, some types have multiple names and definitions, such as histograms and bar charts. We unify these types for simplicity. Second, some types are semantically similar, such as graphs and trees. The tree is a special case of the graph that has a hierarchical data structure. The semantic similarities provide more fine-grained differentiation between classes that help us gain more insights into the composition patterns. For example, graphs are commonly overlaid on map visualizations but trees are not. To avoid duplicated annotations, we only assign one class to each visual component based on its shapes. For example, a visual component

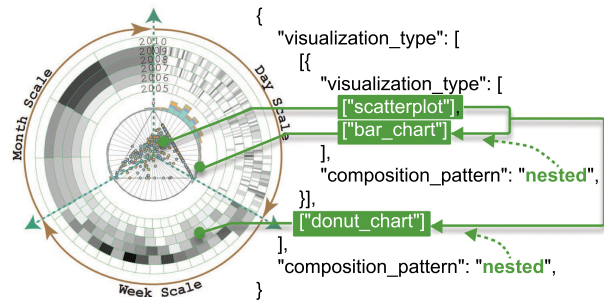


Fig. 2. An example of composite visualization. OpinionSeer [27] is composed of scatterplots, bar charts, and donut charts. The composite visualization can be defined recursively with a hierarchical structure.

is assigned one of the labels “tree” or “graph”. We will choose the label that more accurately describes the visual components. Third, Borkin’s taxonomy does not cover scientific-specific visualizations (e.g., volume rendering), so we add a type named “SciVis”.

- *Composition patterns* refer to the visual relationships between components in a composite visualization. In this work, we propose a taxonomy of composition patterns considering relative positions (e.g., overlapping) and attribute relationships (e.g., type and style) between components. According to the terminology in Javed and Elmqvist [2], where composition patterns describe the usages of space and a relation between the visual components in composite visualization. Therefore, we regard composition patterns as a set of reusable configurations for the generation of a new visualization design given basic visualizations, which demonstrates general visualization design patterns.

Given these two characteristics, we define a *composite visualization* by its components organized with composition patterns, where a component can be either a basic visualization or another composite visualization. Using this recursive definition, a composite visualization can be represented using a hierarchical structure (Fig. 2). Note that for a composite visualization, composition patterns are necessary, which is different from other multiple-view visualizations (or user interfaces) that can be loosely defined as a group of charts placed together [6], [34].

4 CORPUS CONSTRUCTION

This section demonstrates how we constructed the corpus of composite visualizations based on VisImages [26].

4.1 Collecting Figures and Designs

Our main goal is analyzing composite visualizations in research publications of visualizations. Therefore, we constructed the corpus based on VisImages, a dataset that collects figures (as well as the basic visualization types and positions in the figures) from IEEE VIS proceedings. We focused on the papers from 2006 (when VAST was established) to 2020 and obtained 19,910 figures from 1,963 papers. Many other visualization venues, such as EuroVis, ACM CHI, Diagrams, and Infovis journal, also contain high-quality visualization designs. As a starting point, we primarily focus on IEEE VIS and leave the analysis of these venues to future research.

However, the collected figures have purposes that are not suitable for follow-up composite visualization analysis. For example, a large part of the figures is statistical figures used in evaluations, which should be excluded from the corpus. Therefore, we only kept the figures containing original visualization designs. We established three criteria for the filtering. First, we only selected figures containing visualization designs used for data analysis. We excluded figures that illustrate models, frameworks, experiment results, etc. We also excluded figures showing visualization designs from previous papers, such as figures in survey papers [35]. Second, if the components of a design appear in several figures, we selected the one with the most components to maximize the design integrity. Finally, if there are multiple figures duplicated in terms of visualization design, we selected the first one.

We developed an interactive tool for figure selection and design annotation. The tool helps users verify if the figures meet the three criteria and locate the visual designs in the verified figures. Using this tool, three authors independently filter the corpus based on the three criteria. The inconsistent results are discussed and resolved using the majority voting rule. As a result, we filtered 1,353 figures from the 19,910 figures and collected 1,565 visualization designs from the filtered figures.

4.2 Annotating Composite Visualizations

For each composite visualization, we attempted to annotate the composition patterns for further analysis. In VisImages, all basic visualization types and their positions in the figures are identified using the type taxonomy proposed by Borkin et al. [33]. We further verified and revised the visualizations based on the taxonomy descriptions in Fig. 3 with the interactive tool. Thereafter, the visualization designs are decomposed into a series of basic visualization types. Please note that in some cases, a visualization can be assigned with multiple type labels: when the visual component is a heatmap (defined based on color encodings) and other visualization types (based on shapes) simultaneously. However, the multiple label issue is not prevalent (19/1,859). We retain the type heatmap as it can provide information about how composition pattern is used for this particular visualization type. Based on the decomposition, we annotate the composition patterns in a bottom-up manner. First, we analyzed and collected different composition patterns according to the spatial and attribute relationships in each example, then built a taxonomy of composition patterns based on the collected patterns. Second, we labeled all designs in the corpus with the taxonomy and filtered the composite visualizations for the follow-up statistical analysis.

4.2.1 Analyzing Composition Patterns

Cross-referring the taxonomy proposed by Javed and Elmqvist [2] and the collected corpus, we constructed a two-level classification of composition patterns. We term the classification as a taxonomy because we follow a series of rules to exhaustively and exclusively divide the design space into several sub-spaces. At the top level, according to the overlapping relationships between basic visualizations, we identified three patterns, namely, **juxtaposition**, **overlay**, and **nesting**.

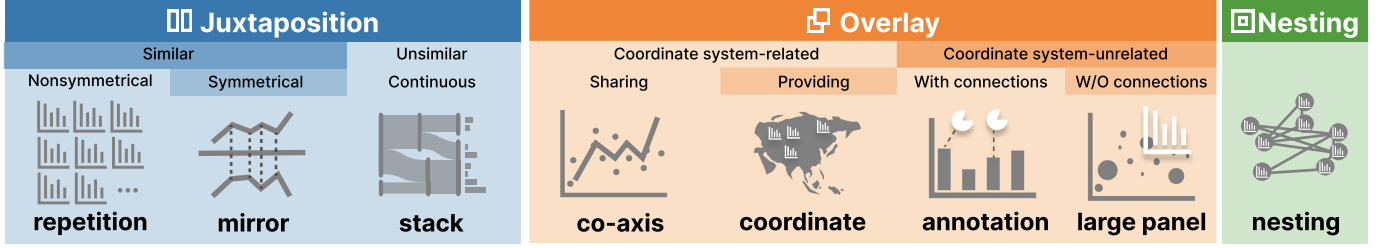


Fig. 3. A two-level taxonomy of Composition Patterns. Composite visualizations are classified according to overlapping relations at the first level, and further classified into sub-types according to geometric (symmetric or continuous) and semantic relation (e.g., sharing coordinate systems or providing coordinate systems for other components).

Compared to the taxonomy proposed by Javed and Elmquist (Fig. 1), ours has several major changes. First, the integrated views are considered as juxtaposition visualizations in our taxonomy, because we consider the explicit visual links and underlying data flow as coordination methods between juxtaposed visualizations. Second, the superimposed and overlaid views are merged as overlay visualizations, because their views are both composed by visually overlaying visualizations on others.

More importantly, for each composition pattern, we further identified several sub-types and contributed a second-level taxonomy based on more fine-grained visual features. The proposed taxonomy and the features to identify each type are presented in Fig. 3. For juxtaposition patterns, we referred to the Gestalt principle and firstly identified visualizations with components of the same visual structures or different visual structures following the similarity rule. For the ones composed of similar visual components, we further identified **repetition patterns** and **mirror patterns** following the symmetry rule, while for the ones with different structures, we identified **stack patterns** based on the continuity rule. For overlaid patterns, we first identified visualizations whose components have coordinate relation and that have no coordinate relation. For the ones whose components have coordinate relation, we identified *co-axis patterns* in which different components share the same coordinate system and *coordinate patterns* in which one component serves as a coordinate system for other components. For the ones whose components do not have coordinate relation, we differentiate *annotation patterns* and *large-panel patterns* based on the existence of connections between the child components and parent components.

4.2.2 Annotating Composition Patterns

With the basic visualizations and composition pattern taxonomy, we further aggregated composition patterns based on different basic visualization types. To ensure the completeness of the analysis, we analyzed the composition patterns by enumerating all possible combinations of the types. For example, imagine a visualization that multiple glyphs of bar charts are distributed on a map, and the map is a heatmap at the same time. We will assign three basic visualization types to the visualization and obtain three visualization type pairs. For each pair of types in a composite visualization, we annotated its composition pattern. Since a composite visualization may have three or more components and is defined recursively, the annotation is also performed in a bottom-up and recursive manner. Similar to

basic visualization types, all composition patterns were annotated and verified by at least two authors. All inconsistencies are resolved by involving a third author and the majority voting rule. In total, we obtained 1,748 composition patterns from 1,859 visualization type pairs.

Please note that, in a composite visualization, the combination of two visualization types is only counted once to avoid redundancy, regardless of the instance numbers of each type. For example, in a scatterplot matrix, we only count the combination of *scatterplot* + *matrix* once.

5 COMPOSITE VISUALIZATIONS IN IEEE VIS

Fig. 4 shows the overall statistics of composition patterns. In this section, we present details for each pattern by reporting the numbers and summarizing their utility.

5.1 Juxtaposition

For juxtaposition visualizations, their components do not overlap and are positioned side by side. Although a user interface (UI) may also consist of visualizations without overlapping, it is not considered a juxtaposition visualization in the context of this study. The component interrelationships in a UI are considerably looser than those in a visualization with juxtaposition patterns. As a rule of thumb, we consider a UI an arbitrary placement of visualizations or interface components (e.g., buttons, sliders, and progress bars) and only extract the visualizations.

Compared to overlay and nesting where visual components are overlaid on or contained by other components, juxtaposition offers a flexible and clear layout for visual components without visual occlusions. Juxtaposed components interrelate through visual links or data flow. Visualization with juxtaposition patterns take up 53.8% (941/1,748) of all composition visualizations, which is the most frequent pattern. We identified three sub-patterns, namely, *repetition*, *mirror*, and *stack*, corresponding to the similarity, symmetry, and continuity rules of the Gestalt principle.

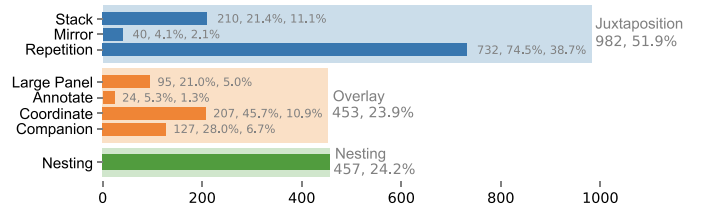


Fig. 4. Distribution of composition patterns.

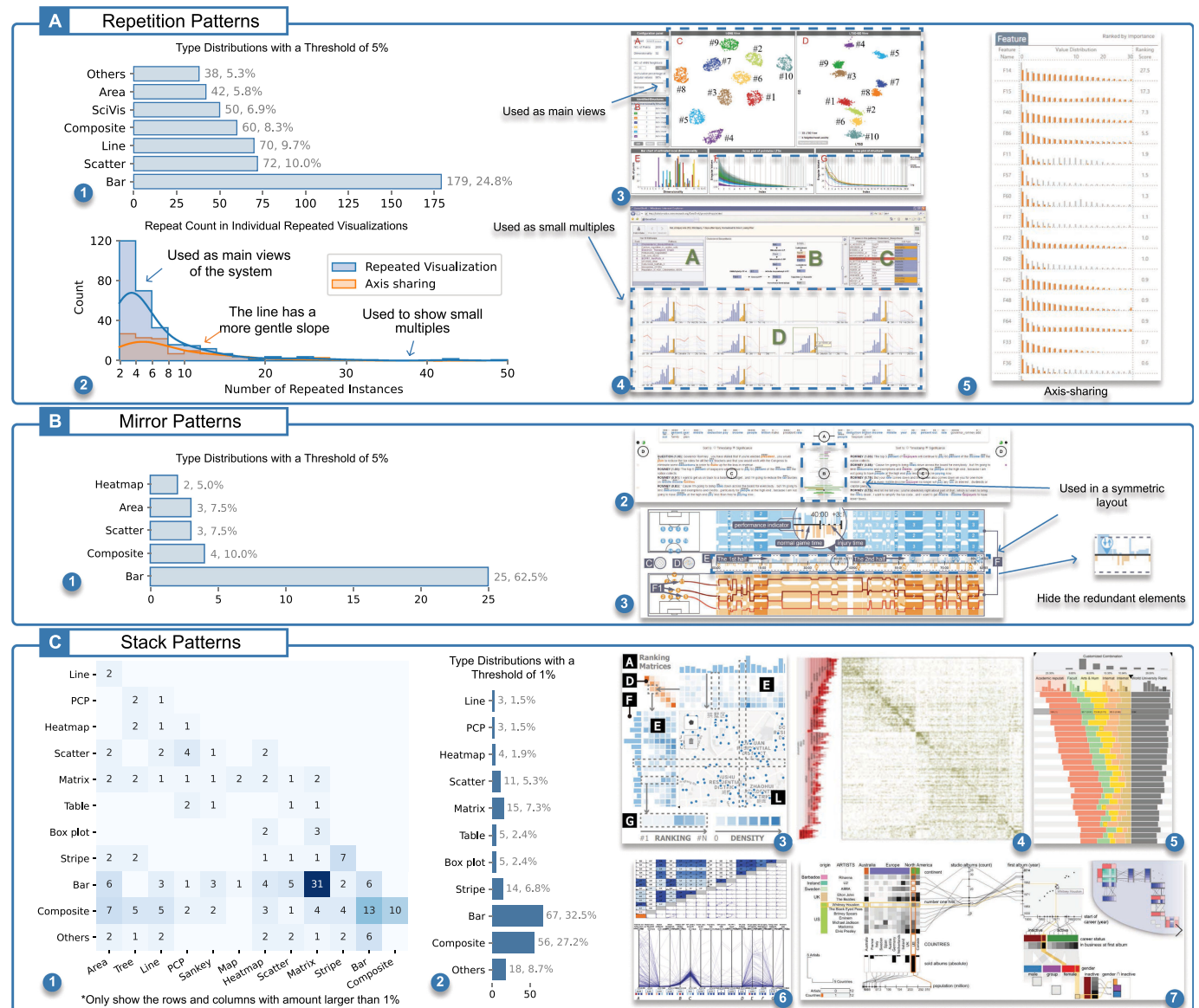


Fig. 5. Statistics and examples of repetition visualizations (A: A3 [36], A4 [37] & A5 [38]), mirror visualizations (B: B2 [39] & B3 [40]), and stack visualizations (C: C3 [41], C4 [42], C5 [43], C6 [44], & C7 [45]).

5.1.1 Repetition Patterns

Definition: repetition refers to the juxtaposition visualizations which are of the same structure (visualization type or composite visualization), but their components are not symmetrical with respect to coordinate axes. Repetition patterns are the most common (41.8%, 730/1,748) among all patterns.

Fig. 5-A1 shows the occurrences of visualization types with repetition patterns. We can see that the distribution is highly skewed. Bar charts (25.1%, 179), scatterplots (10.0%, 72), and line charts (9.7%, 70) are the most frequently used. Besides, composite visualizations (8.3%, 60) and SciVis (6.9%, 50) are also popular with repetition patterns.

Fig. 5-A2 shows the distribution of repetitions in individual repetition visualizations. The long tail distribution indicates two different usage scenarios. On the one hand, some visualizations may contain a large number of components

(up to 100 in our collected examples), as shown in Figs. 5-A4 and 5-A5. These examples generally target lightweight usage, such as browse, selection, or overview, referring to small multiples in other research [46]. The information presented in a repetitive unit is often simple for better readability, and the components are commonly shown in a list view with a scroll bar to handle the scalability. On the other hand, from the peak of the distribution, where the repetition counts lie in [2,4], we discovered a different usage scenario of exploration. For example, in Fig. 5-A3, two scatterplots are used to show data projection with t-SNE and LTSD-GD, respectively [36]. This visualization is used as the main view for showing details of the data, requiring a relatively large space. Among the visualizations with repetition patterns, 38% consist of components that share the same axis. The repetition distribution in this subset also follows a long tail pattern, but with a more gentle slope. With shared axes, repeated components can be easily compared at the same scale.

Repetition patterns have three main advantages. First, the visual similarity of repeated components provides a strong visual hint of grouping according to the Gestalt principle [47]. Second, repetition visualizations can help users in exploring and comparing multiple items, but might not be the best choice compared to mirrored or superimposed layouts [5], [48]. Third, repetition visualizations are easy to implement because codes can be reused conveniently. However, repetition patterns also have disadvantages. First, it is difficult to compare when the number of components gets large because the distance between targeted components might be large [5]. Second, because of the same appearance of components, users may directly compare visual properties, such as size and position, without a careful reference to the scales and attributes of individual components, leading to incorrect insights. In summary, we observe two phenomena in visualizations with repetition patterns:

- Keeping the component number between 2 and 8 in a repetition visualization is the most popular. When using it as the main view for analysis, the number is even fewer (2,4).
- In many examples, it tends to omit unnecessary visual elements to reduce visual clutter. For example, using a shared axis and removing duplicated marks to save space and assist comparison (Fig. 5-A5).

5.1.2 Mirror Patterns



Definition: mirror pattern refers to symmetrically placing two components of the same structure with respect to a coordinate axis. Two components have the same scale on both sides of an axis of symmetry. Visualizations with mirror patterns are much less popular (2.3%, 40/1,748) in our corpus. As shown in Fig. 5-B1, for mirror patterns, the most frequent basic type is bar chart (62.5%, 25). Other popular types used in mirror visualization include composite visualization (10.0%, 4), scatterplot (7.5%, 3), area chart (7.5%, 3), and heatmap (5.0%, 2).

Visualizations with mirror patterns mainly have two advantages. First, taking advantage of people's experience with mirror reflections, mirror visualizations imply identical objects and invite people to compare the two mirrored components [49]. Second, mirror visualizations are aesthetically pleasing because of their symmetry. From the samples, we discover that in some cases [39], [40], mirror visualizations are adopted as a part of a symmetrical design, as shown in Figs. 5-B2 and 5-B3. However, mirror visualizations also have two obvious drawbacks. First, they only support comparing two data series. Second, because of the symmetry layout, it is more difficult to discover precise differences between components. Instead, according to the studies by L'Yi et al. [5], overlaying two data series or using explicit encoding is better than mirror/repetition visualizations in spotting subtle differences. Therefore, mirror visualizations might not be a good choice when the main design goal is precise comparison, but they can be used as auxiliary components within a symmetric design. When using mirror visualizations, a number of designs use an explicit encoding to represent the difference [5], [50] or hide the redundant elements. For example, in ForVizor [40], a soccer analytics

system, when visualizing defensive effectiveness, the bars of the offensive team are hidden (Fig. 5-B3).

5.1.3 Stack Patterns



Definition: components of different types or structures are aligned or concatenated by the same data items or a shared margin (e.g., axis) in a stack visualization. For example, the bar chart + matrix in Fig. 5-C3 and the icicle plot + matrix in Fig. 5-C4 are aligned by items. The shared margin might not be an axis with the same scale, such as Fig. 5-C5, where the bar charts are stacked together with x-axes representing different levels of data. Please note that, although a repetition visualization may also have a shared axis, stack visualizations are different in terms of representation and usage. First, the components in a stack visualization are often different, while components in a repetition visualization are strictly homogeneous. Second, a repetition visualization is mainly used for listing similar data, while a stack visualization focuses more on presenting different facets of the same data in an interconnected manner.

In our corpus, stack visualizations take up 9.8% (171/1,748). From the co-occurrence matrix (Fig. 5-C1), we discovered that the most frequent combination is bar chart + matrix. Among all types, the bar chart is the most frequent (32.5%, 67).

Going through examples, we identified a common usage that bar charts serve as supporting components to another prominent component with a larger size and a more central position. The supporting charts are mainly used to show visual summaries of the main component. For example, a bar chart can summarize the distribution of the data on rows (or columns) of a matrix (Fig. 5-C3). For other stack visualizations, components often have similar sizes and represent different data aspects (Fig. 5-C4), where there are no leading or supporting roles.

Besides, we observed cases where more than two components can be stacked together to form a large visualization. In these cases, some components are used as intermediates to connect two or more components. For example, in Fig. 5-C3, the map connects the heatmaps on the top and to the left. In addition to the grid layout of components in this example, we also observed a novel linear layout (Fig. 5-C7), where matrix, scatterplot, Sankey diagram, and bar chart are connected together through mutual stacking. The visual elements of intermediates handle different aspects of data, such as attributes (e.g., coordinates of parallel coordinate plots), data type (e.g., nominal axis and numeric axis of bar charts), data items (e.g., rows and columns of matrices), and data groups (e.g., nodes of Sankey diagrams). One drawback of linear layout is insufficient space usage when stacking multiple visualizations in different directions (Fig. 5-C8).

Stack visualizations have two major advantages. First, a stack visualization can present different aspects of the same data at the same time in a compact manner, which conforms the rule of space/time resource optimization in designing multiple-view visualizations [51]. Second, relationships between two stacked components are maintained by a shared margin or shared visual items. Users can conveniently switch back and forth between components to explore the data

because of such visual continuity. However, the visual continuity is reduced when the number of items or the distance between aligned items increases. For example, in Fig. 5-C4, although the icicle plot and matrix are adjacent, the icicles with large numbers are distant from the matrix cells, making the alignment and interpretation difficult. We discover the following phenomena based on our observation of stack visualizations:

- A visualization could be created by connecting different charts for better visual coherence using intermediate components if the back-end data of the charts are related (e.g., Fig. 5-C8). In this case, stacking multiple visualizations along the same direction or using a grid layout could improve space usage.
- When the number of aligned items is large or the alignments obscure, visual hints are used to indicate the relationships, such as color encoding, highlighting-on-hover, or visual links. For example, the bar charts in Fig. 5-C5 are not strictly aligned, but the color encoding helps to identify the correspondences.

5.2 Overlay

Visual components are overlaid over other components in an overlay visualizations. Overlay visualizations take up 23.7% (415/1,748) of the collected examples.

Overlay visualizations have two advantages. First, a visualization with overlay patterns often has a more compact layout compared with juxtaposition. Second, overlay patterns can directly represent the correspondences between different components, thus enhancing the visual effect. However, a common disadvantage of overlay patterns is occlusion when compared with juxtaposition visualizations and nesting visualizations. Therefore, when designing an overlay visualization, it would be better to use clutter reduction techniques (e.g., edge bundling) to improve the visual appearance. Overlay visualizations can be organized in four categories: *co-axis*, *coordinate*, *annotation*, and *large panel*. A co-axis visualization contains multiple visualizations that share the same coordinate system, while the other three categories all refer to cases that smaller visualizations (*child components*) are overlaid on the top of larger visualizations (*parent components*).

5.2.1 Co-Axis Patterns



Definition: component visualizations share the same coordinate system in a co-axis visualization. Co-axis visualizations take up 6.9% (121/1,748) in our corpus. Fig. 6-A2 shows that the most frequent type include line chart (43.1%, 53), scatterplot (32.5%, 40), area chart (22.8%, 28), and bar chart (24.4%, 30). From the co-occurrence matrix (Fig. 6-A1), we observe that the top five combinations are bar chart + line chart (15), scatterplot + line chart (14), line chart + area chart (13), box plot + scatterplot (8), and scatterplot + contour graph (6). Going through the samples, we discover several specific tasks for these common combinations. First, scatterplots, which show detailed data items, are often combined with other summary visualizations of trends (line charts), distributions (box plots and contour graphs), etc. The combination of area chart and line chart exhibits various usages, such as showing uncertainty

or differences of the lines with area chart [67], representing aggregated lines with areas [68] to reduce visual clutter, or using area charts as a special case of line charts [69]. The bar chart + line chart is adopted to visualize independent data series in most cases [53], [70], [71], [72]. In rare cases, line charts serve to show density plots for bar charts [73]. In particular, we discovered 4 out of 15 cases in which bar charts and line charts likely share the same coordinate systems, but they actually have different y-coordinates, which would be easily overlooked (e.g., Fig. 6-A4).

The advantage of co-axis patterns is that placing multiple components in the same coordinate system facilitates direct comparison and pattern recognition. We obtain two observations considering the occlusions between components.

- Overlaid components might use transparency to reduce occlusion or put summary/important components on the top. For example, placing box plots on top of a scatterplot for anomaly detection tasks.
- A number of designs adopt multiple coordinate systems in a co-axis visualization, which might introduce potential biases [74]. For example, the bar chart and line chart in Fig. 6-A4 use different y-coordinates, so that users may misinterpret.

5.2.2 Coordinate Patterns



Definition: in a visualization design with coordinate design patterns, parent components provide coordinates (e.g., Cartesian coordinate system, geographic coordinate system, and other reference systems such as grids of the matrix) for child components (or their visual elements). The reference systems are regarded to be part of the parent components. In other words, the positions of child components encode back-end spatial data referring to the parent component. For example, in Fig. 1 B, the map provides spatial context for the graph nodes. Similarly, in Fig. 1 C, the positions of the graph nodes are determined by the treemap grids. Please note that, in some cases, parent components may only provide x- or y-positions for child components. For example, in Fig. 6-B4, the parallel coordinates plot (PCP) provides one of its axes as a reference to the bar chart. Another example is embedding scatterplots into PCP (Fig. 1 D), where one axis is rotated 90 degrees, creating a 2D coordinate system with another axis to host the scatterplots.

In total, we obtained 196 (11.2%/1,748) samples with coordinate patterns. Among the samples, 63 (31.3%) have graphs as child components and 126 (62.7%) contain maps as parents, making the type co-occurrence matrix (Fig. 6-B1) highly sparse. Therefore, we separate them from the type co-occurrence matrix and visualize their type distributions independently (Figs. 6-B2 and 6-B3). Among all combinations, overlaying graphs on maps is the most frequent.

Graphs are mostly used as child components (31.3%, 63). It is likely because, in many graph visualization tasks, analysts mainly focus on topological patterns, such as node degree and node connectivity [75]. Therefore, node positions are relatively flexible depending on the needs of different tasks, and encoding spatial information with node positions becomes a popular design pattern. In addition, there are many cases with word clouds referring to area

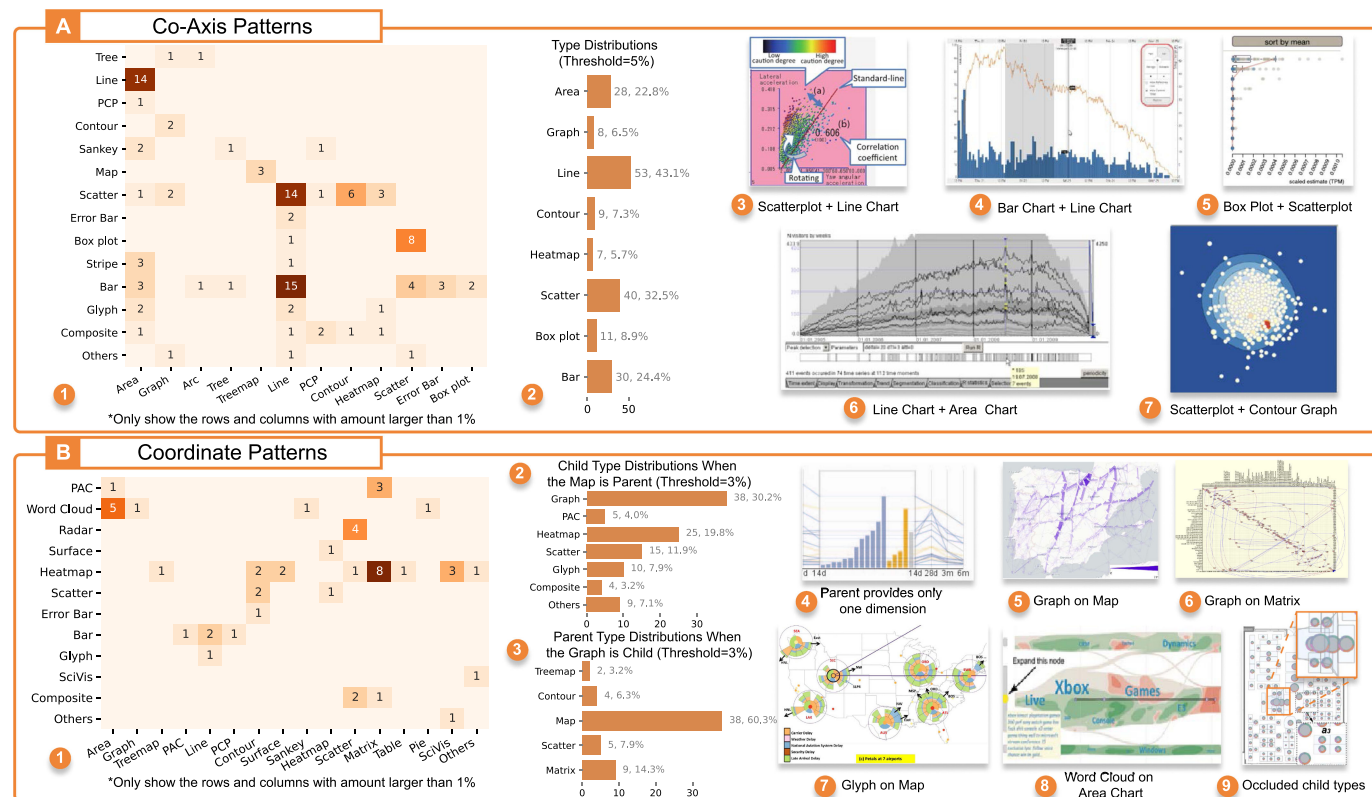


Fig. 6. Statistics and examples of co-axis patterns (A: A3 [52], A4 [53], A5 [54], A6 [55], & A7 [56]) and coordinate patterns (B: B4 [37], B5 [57], B6 [58], B7 [59], B8 [60], & B9 [61]).

charts (Fig. 6-B8) or proportional area charts referring to matrices. In these cases, child components regularly do not present spatial information.

Compared with co-axis patterns where the components have independent but identical coordinate systems, the layout of child components is determined by their parent components in coordinate patterns. Therefore, they are effective in helping users interpret child components in the context of a parent component. We discover two phenomena with coordinate patterns.

- Various designs choose to combine a parent component that provides spatial context and child components whose visual elements do not encode spatial information, such as word clouds and proportional area charts. For example, in Fig. 6-B8, the words in the word clouds are distributed on an area chart to visualize the topic frequency.
- In addition to the inevitable occlusions between child and parent components, the overlapping between the child components may exacerbate the overall occlusions. For example, in Fig. 6-B9, the glyphs are used to enhance the map visualization, but users might fail to retrieve the information encoded by glyphs because of occlusion.

5.2.3 Annotation Patterns

Definition: child components of small size are overlaid on parent components and connected to elements of parent components in annotation visualizations, but the positions of child components do not encode spatial information.

Moreover, the child components provide a “cut-out” lens for the visual elements connected [76]. Compared with co-axis and coordinate, in which components are related to each other because of sharing or providing coordinate systems, annotation visualizations are more flexible as child components have more freedom in placement and can use visual links to explicitly connect to the parent component. There are only 22 (1.3%) annotation visualizations in our corpus.

We reviewed the captions and text descriptions in the corresponding papers to understand the scenarios of annotation visualizations. 17 out of 22 have mentioned that the child components are displayed on demand (via interactions with the parent components). Therefore, the most common usage of annotation patterns is showing additional information with tooltips [56], [77], [78]. For example, in Fig. 7-C2, when hovering on a grid on the map, a graphical annotation about the profile of that grid will present.

The advantage of annotations is the flexibility in positioning child components. We discover the following two phenomena of annotation visualizations.

- In most cases, only details of focused data items are visualized following the rule of details on demand [79].
- The layout of child components can be optimized (e.g., saliency-based method [80]) to utilize empty space or reduce line crossings.

5.2.4 Large Panel Patterns

Definition: child components of small size overlay directly on parent components without visual links in a large panel visualization, and the positions of child components

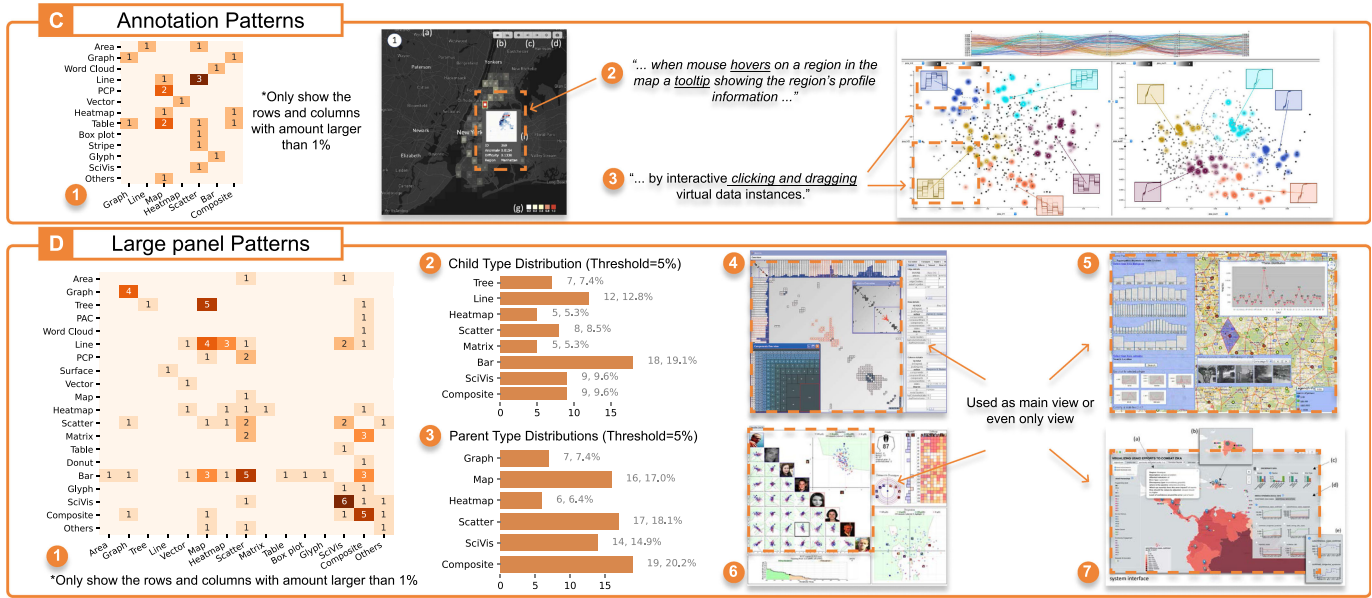


Fig. 7. Statistics and examples of annotation patterns (C: C2 [56] & C3 [62]) and large panels (D: D4 [63], D5 [64], D6 [65], & D7 [66]).

do not encode spatial information. Unlike annotations, large panels do not connect the child and parent components using links or anchors, and the child components show details of the parent components in an overview + detail manner [76].

In total, We obtained 76 (4.3%) samples of large-panel visualizations. Figs. 7-D2 and 7-D3 show the distributions of visualization types used as child and parent components, respectively. By exploring the samples, we found that large-panel visualizations are mostly used as the main views in visual analytics systems, even as the only view in some systems (Fig. 7 D). Child components generally serve as auxiliary views for the whole parent components, not specific elements of the parent components. This feature makes large-panel visualizations different from annotation visualizations, where child components usually present the details of elements in parent components.

Compared with annotation patterns, large-panel visualizations offer more flexibility for placing child components, since they do not require anchoring points in the parent component. For large-panel visualizations, they generally place child components at positions where elements are less important (such as corners) to mitigate visual occlusion.

5.3 Nesting

Definition: in nesting visualizations, some components (denoted by **child components**) are embedded into the visual elements or internal area of other components (denoted by **parent components**). We collected 392 (22.4%) samples of nesting visualizations in total. The co-occurrence matrix (Fig. 8-A1) shows that nesting visualizations have more diverse type combinations than other composition patterns. The most frequent combinations include scatterplots + matrix (a.k.a scatterplot matrix) and bar chart + table. Bar charts and scatterplots are the most common child type. For parent components, graphs, matrices, tables, Sankey diagrams, bar charts, parallel coordinate plots (PCP), and scatterplots are significantly more popular than other types. We also observed different patterns of visualizing child components in nesting visualizations. Small elements of parent components, such as nodes of graphs (Fig. 8-A5), nodes and flows of Sankey diagrams (Fig. 8-A4), sectors of donut chart, and cells of matrix and table are obvious visual spaces to embed child components. However, some other parent visual elements, such as axes of PCP, need to distort to create a canvas to host child components

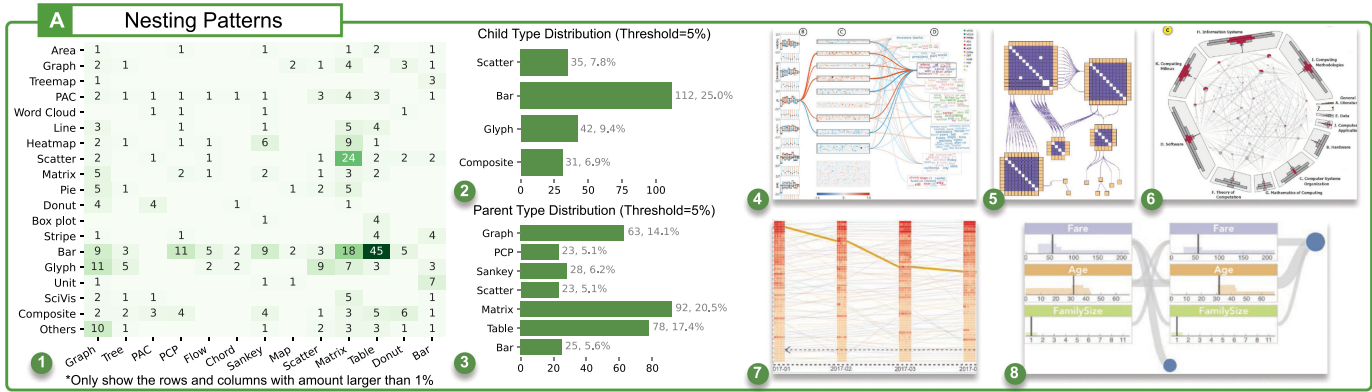


Fig. 8. Statistics and examples of nesting visualizations (A4 [81], A5 [82], A6 [83], A7 [84], A8 [85]).
Authorized licensed use limited to: Zhejiang University. Downloaded on December 08, 2023 at 07:18:15 UTC from IEEE Xplore. Restrictions apply.

(Fig. 8-A7). In addition, there are nesting visualizations where parent components have circular shapes and internal area, such as donut (Figs. 2 & 8-A6). The internal area can provide relatively sufficient space other than visual elements. These cases are not common (6.4%, 25) among nesting visualizations.

Nesting visualizations have two advantages. First, they have no occlusions between parent and child components and imply hierarchical information, compared to overlay visualizations. Therefore, nesting visualizations can visualize the overview of parent components (e.g., the overall layout of graphs) while maintaining details of the child items (e.g., graph nodes and matrix cells) [86]. Second, nesting visualizations are more compact than overlay visualizations and juxtaposition visualizations. However, one major limitation of nesting visualizations is the limited space of visual elements to host child components. From the observations, we discover two phenomena.

- A number of designs choose to use relatively common/simple visualizations in child components, such as bar charts (Fig. 8-A8) and heatmaps (Figs. 8-A4 and 8-A7). We infer that this is because visualizations with complex configurations are hard to identify due to the limited space of child components.
- A number of designs apply geometric transformations to the elements of parent components to make room for child components. For example, Sun et al. [87] proposed a route-zooming technique to distort the map for hosting visualizations for spatio-temporal information.

6 USAGE SCENARIO

The taxonomy and corpus can be used in different aspects.

6.1 Exploring & Understanding Visual Designs

The corpus can help researchers and designers in explore and understand composite visualization designs. To facilitate design exploration, we developed an explorer for composite visualizations. The explorer consists of four components: a **filtering panel** (Fig. 9-A), a **heatmap view** (Fig. 9-B), a **gallery view** (Fig. 9-C), and a **detail view** (Fig. 9-D). The filtering panel supports filtering designs by keywords, year, venue, composition pattern, and visualization type. After filtering, an overview of type combinations under different composition patterns is displayed in the heatmap view. Each grid in the heatmap encodes the number of a type combination. The heatmaps support filtering designs of specific type combinations by clicking on the corresponding cells. The filtered designs are displayed in the gallery view. When clicking on a design in the gallery view, a window pops up to show details about that design, including our annotations and metadata, such as the title. The interface facilitates searching for a visualization type with different composition patterns. For example, Fig. 9-D1 shows bar charts being displayed on a map visualization with the coordinate pattern, and Fig. 9-D2 nests bar visualizations into the graph nodes.

Moreover, using the interface to display visualizations of specific type combinations and composition patterns side

by side enables understanding the evolution of the design. For example, we can witness an evolution of design complexity for the visualizations with stacking bar charts and matrices from Fig. 9 (Right). In earlier years (2006 and 2008), designs differentiate from each other in terms of shape and color encodings of the matrix cells. Then more different visual elements are concatenated with bar charts, such as boxplots (2014). Composite visualizations of bar charts and matrices can further serve as support components for map visualization (2018). Besides, there are different directions for matrices when stacking with bar charts. In a more recent design (2020), bar charts are stacked with multiple matrices in a crossing/exploding layout.

6.2 Training Data for AI4VIS

Our corpus can be used as training data for artificial intelligence models for visualizations (AI4VIS) [88]. A possible task is decomposition, which comprises two sub-tasks, i.e., recognizing positions and types of basic visualizations and inferring the composition patterns of the visualizations. The bounding boxes and labels can be used for visualization detection [25], [26]. For the visualizations assigned with multiple labels (e.g., heatmap and map), the data is represented with multiple bounding boxes with the same x, y, width, and height but different labels. VisImages [26] has demonstrated a case for this situation. With bounding boxes and visualization types, we can further infer the composition patterns, which describe the relationships between the basic visualizations. Recent studies propose practical methods to learn the hierarchical structures of visual elements with graph neural networks [89] or transformer-based model [90]. Furthermore, our annotation comprises the co-occurrence between basic visualizations, which could be used for visualization recommendation [91] based on knowledge graph (e.g., basic visualizations as the entities and design patterns as the relations).

7 FUTURE RESEARCH OPPORTUNITIES

In this section, we discuss future research opportunities for implementing composite visualizations and exploiting empirical evidence for task-driven efficiency.

7.1 Implementing Composite Visualizations

Our taxonomy can be used to measure the expressiveness of existing visualization grammar in visualization rendering and facilitate the development of more ease-of-use visualization generation grammar. Grammars that support operating on low-level visual elements, such as D3.js [92] and Vega [93], can implement various composition patterns through programming, but they require high programming capability of the users.

In recent years, declarative programming languages are developing rapidly and gradually supporting the generation of visualizations through the intuitive specification of the visual encodings. For example, Vega-Lite [18] supports view composition with operators of “facet/repeat” (repetition), “layer” (part of coordinate patterns and co-axis patterns), and “vconcat/hconcat” (part of stack patterns). However, it cannot generate nesting visualizations when there is a need to represent additional information in the visual elements,

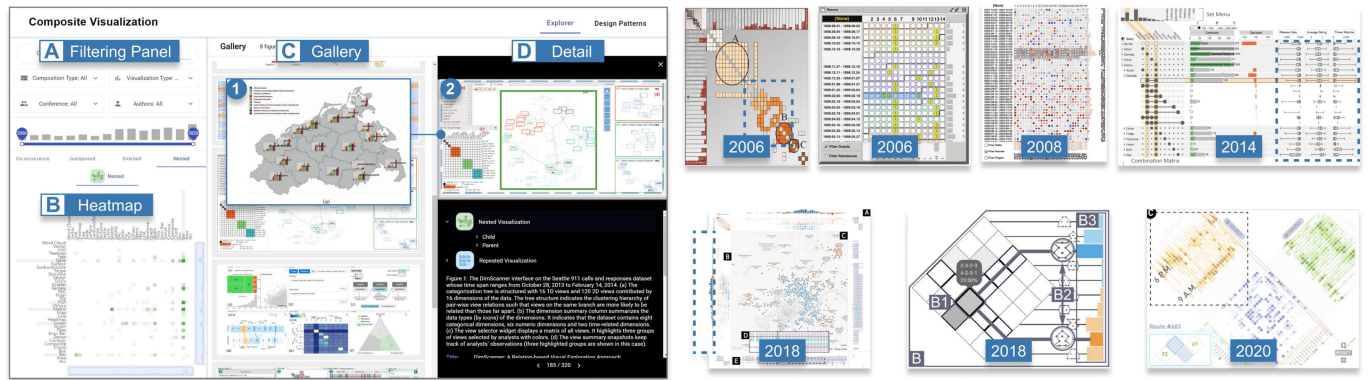


Fig. 9. Left: composite visualization explorer with a filtering panel (A), a heatmap view (B), a gallery (C), and a detail view (D). Right: stack visualizations with matrices and bar charts across different years.

such as glyph visualizations [94]. The generation of nesting visualizations also requires the support of processing network/hierarchical data. ECharts [95] supports graph and tree visualizations. GoTree [96] facilitates the rendering of hierarchical data by specifying coordinate systems, visual elements, layout, etc. However, these grammars lack original support for composition. ATOM [97] supports generating visualizations with nested data units, but the building blocks of generated visualizations are not visualizations. Nevertheless, its graphical operations, such as bin, duplicate, and filter, can be extended to support generating nesting visualizations, for example, aggregating the transformed data units and rendering them with basic visualization types. In all, analyzing existing visualization grammars with the taxonomy, we understand that existing grammars can be further extended to support the convenient generation of more composition patterns, especially nesting patterns, which account for 24.2% in our corpus.

7.2 Empirical Evidence for Task-Driven Efficiency

Composition patterns have been applied in many visual analytics systems, which are designed to achieve various analysis tasks [98]. However, the efficiency of visualization composition has long been discussed and the design complexity remains a problem in visualization research [99], [100], [101]. However, directly comparing the design of different visual analytics systems might be impractical due to their complexity. Our taxonomy provides a breakdown of composition for evaluating the efficiency of complex designs under different tasks [98], [102], such as comparing values and discovering anomalies.

Some studies have investigated composite visualizations for the task of comparison. For example, Isenberg et al. [74] studied how dual-axis charts, a special co-axis pattern, perform in the tasks of comparing lengths and distances. L'Yi et al. [5] have thoroughly explored the effectiveness of repeated, mirrored, and co-axis layouts for comparison.

In addition, we observe the use of composite visualizations for other tasks, such as co-axis patterns for discovering correlations/anomalies and coordinate patterns for providing spatial information. Saket et al. [103] have studied how basic visualization types perform in low-level tasks (e.g., finding anomalies, finding clusters, and correlation) [102]. However, few studies have investigated the efficiency

of composition patterns for these tasks. Future studies can conduct controlled experiments in exploring the efficiency of several representative visualizations with compositions.

Nesting visualizations are well suited for representing the network and hierarchical data [86]. Still, these data are usually analyzed under tasks different from tabular data, such as perceiving the topology and the attributes on nodes or links [75]. Existing studies have investigated the perception efficiency of graphs in different conditions, such as static or dynamic manners [104] and multiple sampling models [105]. However, few user studies have been conducted to evaluate the efficiency of nesting layouts. Future studies can design different perception tasks for the child components and parent components in the visualizations and conduct controlled experiments accordingly.

8 DISCUSSION

In this section, we discuss trade-offs of different composition patterns and limitations of composition representations.

8.1 Balancing Expressiveness and Effectiveness

Designing a visualization should handle the trade-offs between representing more information in a limited visual space and ensuring users are not overwhelmed by too many visual components [1]. Specifically, juxtaposition can provide flexible layouts for charts with small occlusions, which might be friendly for design novices. Among juxtapositions, stack patterns can express different aspects of data with a more coherent arrangement. Furthermore, overlay and nesting provide more compact layouts than juxtaposition visualizations to handle more visual components at once. These compositions can also make convenient to perceive spatial, networking, or hierarchical relation between child and parent components. However, these patterns increase the visual occlusion and limit the size of child components.

8.2 Coverage of Composition Representations

We encountered that some designs are not decomposable with our composition representations.

For example, Bubble Sets [106] are highly customized with primitive shapes (such as rectangles, lines, and circles),

instead of combining multiple types together. Moreover, the composition patterns might not fully reflect the design novelty. For example, in addition to coordinate and nesting, Whisper [107] use a metaphor of sunflower to represent the retweeting activities.

To understand and analyze these novel designs, lower-level decomposition, which concerns transformations and visual encodings, is required, such as the component layout (e.g., circular and branched).

8.3 Limitations

Our study has two limitations. First, data and interactions of designs, which are closely related to analytical tasks and design requirements, are not considered in our corpus. The acquisition of this information requires extensive efforts on paper reading and is even inaccessible sometimes. In this work, we mainly study design patterns regarding observable information in figures, such as visualization types and spatial relationships, because of the large size of the corpus. A potential solution to retrieve information about data and interactions might be using natural language processing techniques to extract and analyze related text descriptions.

Second, the corpus construction mainly relies on manual annotation, which is limited in scalability for a larger quantity of data. Recently, studies [25], [26] have adopted object detection models to process visualization images, which might be a promising method for data collection. In this work, we lacked a well-curated dataset for model training at the beginning, but we could use the collected corpus to explore the potential of automatically recognizing visualizations and composition patterns.

9 CONCLUSION AND FUTURE WORKS

In this work, we opted to answer the question of what and how visualizations can be composed together to form novel designs. To achieve this, we conducted a demographic analysis on composite visualizations, based on a corpus of visual designs from IEEE VIS. With the corpus, we proposed a taxonomy of eight design patterns. For each design pattern, we analyzed the distributions and correlations between different visualization types, and obtained insights on usage scenarios, advantages, disadvantages, and design suggestions.

We released the corpus and an explorer to advance the studies in designing composite visualizations: <https://composite-visualizations.github.io/>.

For future research, one promising direction is a library that can flexibly integrate different visualization types with different composition patterns. Existing libraries (e.g., Vega-lite, ECharts) can provide support for layering or faceting. Nevertheless, researchers create composite visualizations mainly by writing codes with programming languages (e.g., Javascript), since composite visualizations are with complex structures and are commonly integrated with visual analytics systems. With the taxonomy and corpus obtained in this work, we might extend the features of existing libraries for better creation of composite visualizations.

Another research problem is the effectiveness of composition patterns on different tasks. With the overview provided before, we understand the scenarios and tasks for

different design patterns, but in-depth inspections, such as controlled studies, are beyond the scope of this paper. Recently, L'Yi et al. [5] have explored the effectiveness of different chart compositions (e.g., superimposed and explicit encoding) on visual comparison. We hope our taxonomy and findings could shed light on exploring further empirical studies with more diverse composition patterns, visualizations, and tasks. Furthermore, we could study the interaction techniques in composite visualizations. Specifically, we could use natural language processing approaches with existing interaction taxonomies [108] to retrieve a corpus of interactions in the designs. Combining with the corpus in this work, we might discover the relations between interactions, visualization types, and composition patterns in the works of visualization community.

REFERENCES

- [1] T. Munzner, *Visualization Analysis and Design*. Boca Raton, FL, USA: CRC Press, 2014.
- [2] W. Javed and N. Elmqvist, "Exploring the design space of composite visualization," in *Proc. IEEE Pacific Vis. Symp.*, 2012, pp. 1–8.
- [3] R. Keller, C. M. Eckert, and P. J. Clarkson, "Matrices or node-link diagrams: Which visual representation is better for visualising connectivity models?," *Informat. Vis.*, vol. 5, no. 1, pp. 62–76, 2006.
- [4] R. S. Laramée, H. Hauser, H. Doleisch, B. Vrolijk, F. H. Post, and D. Weiskopf, "The state of the art in flow visualization: Dense and texture-based techniques," *Comput. Graph. Forum*, vol. 23, no. 2, pp. 203–221, 2004.
- [5] S. L'Yi, J. Jo, and J. Seo, "Comparative layouts revisited: Design space guidelines and future directions," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 1525–1535, Feb. 2021.
- [6] X. Chen, W. Zeng, Y. Lin, H. M. Al-Manee, J. Roberts, and R. Chang, "Composition and configuration patterns in multiple-view visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 1514–1524, Feb. 2021.
- [7] Y. Wang et al., "DataShot: Automatic generation of fact sheets from tabular data," *IEEE Trans. Vis. Comput. Graph.*, vol. 26, no. 1, pp. 895–905, Jan. 2020.
- [8] D. Shi, X. Xu, F. Sun, Y. Shi, and N. Cao, "Calliope: Automatic visual data story generation from a spreadsheet," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 453–463, Feb. 2021.
- [9] M. Gleicher, "Considerations for visualizing comparison," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 413–423, Jan. 2018.
- [10] F. Beck, M. Burch, S. Diehl, and D. Weiskopf, "The state of the art in visualizing dynamic graphs," in *Proc. EuroVis State-of-the-Art Rep.*, 2014, pp. 83–103.
- [11] C. Nobre, M. Meyer, M. Streit, and A. Lex, "The state of the art in visualizing multivariate networks," *Comput. Graph. Forum*, vol. 38, no. 3, pp. 807–832, 2019.
- [12] S. Schöttler, Y. Yang, H. Pfister, and B. Bach, "Visualizing and interacting with geospatial networks: A survey and design space," *Comput. Graph. Forum*, vol. 40, pp. 5–33, 2021.
- [13] A. Blackwell and Y. Engelhardt, "A meta-taxonomy for diagram research," in *Diagrammatic Representation and Reasoning*, Berlin, Germany: Springer, 2002, pp. 47–64.
- [14] J. von Engelhardt, *Lists of Illustrations, Maps, Charts, and Diagrams*. Princeton, NJ, USA: Princeton Univ. Press, 2002.
- [15] Y. Engelhardt and C. Richards, "A framework for analyzing and designing diagrams and Graphs," in *Proc. Int. Conf. Theory Application Diagrams*, 2018, pp. 201–209.
- [16] K. Sedig and P. Parsons, *Design of Visualizations for Human-Information Interaction: A Pattern-Based Framework*, Berlin, Germany: Springer, 2016.
- [17] L. Wilkinson, "The grammar of Graph.: The GGPlot2 package," in *Handbook of Computational Statistics*, Berlin, Germany: Springer, 2012, pp. 375–414.
- [18] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A grammar of interactive Graphs," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 341–350, Jan. 2017.

- [19] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu, "Scattering points in parallel coordinates," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1001–1008, Nov./Dec. 2009.
- [20] J. Fekete, D. Wang, N. Dang, A. Aris, and C. Plaisant, "Interactive Poster: Overlaying graph links on treemaps," in *Proc. IEEE Symp. Informat. Vis. Conf. Compendium*, 2003, pp. 82–83.
- [21] I. Lokuge and S. Ishizaki, "GeoSpace: An interactive visualization system for exploring complex information spaces," in *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 1995, pp. 409–414.
- [22] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, "Visual comparison for information visualization," *Informat. Vis.*, vol. 10, no. 4, pp. 289–309, 2011.
- [23] R. Li and J. Chen, "Toward a deep understanding of what makes a scientific visualization memorable," in *Proc. IEEE Sci. Vis. Conf.*, 2018, pp. 26–31.
- [24] W. Zeng, A. Dong, X. Chen, and Z.-L. Cheng, "VISTory: Interactive storyboard for exploring visual information in scientific publications," *J. Vis.*, vol. 24, no. 1, pp. 69–84, 2021.
- [25] J. Chen et al., "VIS30K: A collection of figures and tables from IEEE visualization conference publications," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 9, pp. 3826–3833, Sep. 2021.
- [26] D. Deng et al., "VisImages: A fine-grained expert-annotated visualization dataset," *IEEE Trans. Vis. Comput. Graph.*, to be published, doi: [10.1109/TVCG.2022.3155440](https://doi.org/10.1109/TVCG.2022.3155440).
- [27] Y. Wu et al., "OpinionSeer: Interactive visualization of hotel customer feedback," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 1109–1118, Nov./Dec. 2010.
- [28] G. L. Lohse, K. Biolsi, N. Walker, and H. H. Rueter, "Classification with invariant scattering representations," *Commun. ACM*, vol. 37, no. 12, pp. 36–50, 1994.
- [29] E. H.-H. Chi, "A taxonomy of visualization techniques using the data state reference model," in *Proc. IEEE Symp. Informat. Vis.*, 2000, pp. 69–75.
- [30] R. L. Harris, *Information Graph: A Comprehensive Illustrated Reference*, London, U.K.: Oxford Univ. Press, USA, 1999.
- [31] J. Heer, M. Bostock, and V. Ogievetsky, "A tour through the visualization zoo," *Commun. ACM*, vol. 53, no. 6, pp. 59–67, 2010.
- [32] I. Meirelles, *Design for Information: An Introduction to the Histories, Theories, and Best Practices behind Effective Visualizations*. Beverly, MA, USA: Rockport Publishers, 2013.
- [33] M. A. Borkin et al., "What makes a visualization memorable?," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2306–2315, Dec. 2013.
- [34] J. C. Roberts, "State of the art: Coordinated multiple views in exploratory visualization," in *Proc. Int. Conf. Coordinated Mult. Views Explor. Vis.*, 2007, pp. 61–71.
- [35] J. Johansson and C. Forsell, "Evaluation of parallel coordinates: Overview categorization and guidelines for future research," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 579–588, Jan. 2016.
- [36] J. Xia et al., "LDSScanner: Exploratory analysis of low-dimensional structures in high-dimensional datasets," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 236–245, Jan. 2018.
- [37] B. Kim, B. Lee, S. Knoblach, E. Hoffman, and J. Seo, "GeneShelf: A web-based visual interface for large gene expression time-series data repositories," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 905–912, Nov./Dec. 2009.
- [38] S. Liu, J. Xiao, J. Liu, X. Wang, J. Wu, and J. Zhu, "Visual diagnosis of tree boosting methods," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 163–173, Jan. 2018.
- [39] M. El-Assady, R. Sevastjanova, F. Sperrle, D. Keim, and C. Collins, "Progressive learning of topic modeling parameters: A visual analytics framework," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 382–391, Jan. 2018.
- [40] Y. Wu et al., "ForVizor: Visualizing spatio-temporal team formations in soccer," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 65–75, Jan. 2019.
- [41] D. Weng, R. Chen, Z. Deng, F. Wu, J. Chen, and Y. Wu, "SRVis: Towards better spatial integration in ranking visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 459–469, Jan. 2019.
- [42] A. Bilal, A. Jourabloo, M. Ye, X. Liu, and L. Ren, "Do convolutional neural networks learn class hierarchy?," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 152–162, Jan. 2018.
- [43] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit, "LineUp: Visual analysis of multi-attribute rankings," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2277–2286, Dec. 2013.
- [44] S. Johansson and J. Johansson, "Interactive dimensionality reduction through user-defined combinations of quality metrics," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 993–1000, Nov./Dec. 2009.
- [45] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit, "Domino: Extracting comparing and manipulating subsets across multiple tabular datasets," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 2023–2032, Dec. 2014.
- [46] S. van den Elzen and J. J. van Wijk, "Small multiples large singles: A new approach for visual data exploration," *Comput. Graph. Forum*, vol. 32, no. 3pt2, pp. 191–200, 2013.
- [47] R. Rosenholtz, N. R. Twarog, N. Schinkel-Bielefeld, and M. Wattenberg, "An intuitive model of perceptual grouping for HCI design," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2009, pp. 1331–1340.
- [48] B. Ondov, N. Jardine, N. Elmqvist, and S. Franconeri, "Face to Face: Evaluating visual comparison," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 861–871, Jan. 2019.
- [49] E. R. Tufte, "The visual display of quantitative information," *J. Healthcare Qual.*, vol. 7, no. 3, pp. 1–15, 1985.
- [50] A. Srinivasan, M. Brehmer, B. Lee, and S. M. Drucker, "What's the difference? Evaluating variations of multi-series bar charts for visual comparison tasks," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2018, pp. 1–12.
- [51] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky, "Guidelines for using multiple views in information visualization," in *Proc. Work. Conf. Adv. Vis. Interfaces*, 2000, pp. 110–119.
- [52] M. Itoh, D. Yokoyama, M. Toyoda, and M. Kitsuregawa, "A system for visual exploration of caution spots from vehicle recorder data," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol.*, 2015, pp. 199–200.
- [53] J. Jo, J. Huh, J. Park, B. Kim, and J. Seo, "LiveGantt: Interactively visualizing a large manufacturing schedule," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 2329–2338, Dec. 2014.
- [54] H. Strobelt et al., "Vials: Visualizing alternative splicing of genes," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 399–408, Jan. 2016.
- [55] G. Andrienko, N. Andrienko, M. Mladenov, M. Mock, and C. Pölit, "Discovering bits of place histories from people's activity traces," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, 2010, pp. 59–66.
- [56] N. Cao, C. Lin, Q. Zhu, Y. Lin, X. Teng, and X. Wen, "Voila: Visual anomaly detection and monitoring with streaming spatio-temporal data," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 23–33, Jan. 2018.
- [57] N. Andrienko, G. Andrienko, J. M. C. Garcia, and D. Scarlatti, "Analysis of flight variability: A systematic approach," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 54–64, Jan. 2019.
- [58] D. H. Jeong, S. Ji, W. Ribarsky, and R. Chang, "A state transition approach to understanding users' interactions," in *Proc. IEEE Conf. Vis. Analytics Sci. Technol.*, 2011, pp. 285–286.
- [59] S. Ko et al., "Analyzing high-dimensional multivariate network links with integrated anomaly detection highlighting and exploration," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol.*, 2014, pp. 83–92.
- [60] Y. Wu, S. Liu, K. Yan, M. Liu, and F. Wu, "OpinionFlow: Visual analysis of opinion diffusion on social media," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 1763–1772, Dec. 2014.
- [61] D. Liu, P. Xu, and L. Ren, "TPFlow: Progressive partition and multidimensional pattern extraction for large-scale spatio-temporal data analysis," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 1–11, Jan. 2019.
- [62] D. Orban, D. F. Keefe, A. Biswas, J. Ahrens, and D. Rogers, "Drag and Track: A direct manipulation interface for contextualizing data instances within a continuous parameter space," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 256–266, Jan. 2019.
- [63] N. Henry and J. Fekete, "MatrixExplorer: A dual-representation system to explore social networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 677–684, Sep./Oct. 2006.
- [64] I. Peca, H. Zhi, K. Vrotsou, N. Andrienko, and G. Andrienko, "KD-photomap: Exploring photographs in space and time," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol.*, 2011, pp. 291–292.
- [65] M. A. Migut, J. C. van Gemert, and M. Worring, "Interactive decision making using dissimilarity to visually represented prototypes," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol.*, 2011, pp. 141–149.
- [66] N. Mccurdy, J. Gerdes, and M. Meyer, "A framework for externalizing implicit error using visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 925–935, Jan. 2019.

- [67] M. C. Hao, H. Janetzko, R. K. Sharma, U. Dayal, D. A. Keim, and M. Castellanos, "Poster: Visual prediction of time series," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, 2009, pp. 229–230.
- [68] P. Xu, H. Mei, L. Ren, and W. Chen, "ViDX: Visual diagnostics of assembly line performance in smart factories," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 291–300, Jan. 2017.
- [69] J. Wang, L. Gou, H. W. Shen, and H. Yang, "DQNViz: A visual analytics approach to understand deep Q-networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 288–298, Jan. 2019.
- [70] Q. Li, P. Xu, and H. Qu, "FPSSeer: Visual analysis of game frame rate data," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol.*, 2015, pp. 73–80.
- [71] G. Li, J. Wang, H. W. Shen, K. Chen, G. Shan, and Z. Lu, "CNNPruner: Pruning convolutional neural networks with visual analytics," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 1364–1373, Feb. 2021.
- [72] Z. Zhong et al., "SilkViser: A visual explorer of blockchain-based cryptocurrency transaction data," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol.*, 2020, pp. 95–106.
- [73] S. Chen et al., "Interactive visual discovering of movement patterns from sparsely sampled geo-tagged social media data," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 270–279, Jan. 2016.
- [74] P. Isenberg, A. Bezerianos, P. Dragicevic, and J.-D. Fekete, "A study on dual-scale data charts," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2469–2478, Dec. 2011.
- [75] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry, "Task taxonomy for graph visualization," in *Proc. AVI Workshop Beyond Time Errors Novel Eval. Methods Informat. Vis.*, 2006, pp. 1–5.
- [76] B. Bach, Z. Wang, M. Farinella, D. Murray-Rust, and N. Henry Riche, "Design patterns for data comics," in *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2018, pp. 1–12.
- [77] S. G. Eick, M. A. Eick, J. Fugitt, B. Horst, M. Khailo, and R. A. Lankenau, "Thin client visualization," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, 2007, pp. 51–58.
- [78] S. Guo, K. Xu, R. Zhao, D. Gotz, H. Zha, and N. Cao, "EventThread: Visual summarization and stage analysis of event sequence data," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 56–65, Jan. 2018.
- [79] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proc. IEEE Symp. Vis. Lang.*, 1996, pp. 336–343.
- [80] M. Steinberger, M. Waldner, M. Streit, A. Lex, and D. Schmalstieg, "Context-preserving visual links," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2249–2258, Dec. 2011.
- [81] Y. Ming et al., "Understanding hidden memories of recurrent neural networks," in *Proc. IEEE Conf. Vis. Anal. Sci. Technol.*, 2017, pp. 13–24.
- [82] N. Henry, A. Bezerianos, and J. Fekete, "Improving the readability of clustered social networks using node duplication," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1317–1324, Nov./Dec. 2008.
- [83] B. Alsallakh, W. Aigner, S. Miksch, and H. Hauser, "Radial Sets: Interactive visual analysis of large overlapping sets," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2496–2505, Dec. 2013.
- [84] X. Yue et al., "BitExTract: Interactive visualization for extracting bitcoin exchange intelligence," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 162–171, Jan. 2019.
- [85] X. Zhao, Y. Wu, D. L. Lee, and W. Cui, "iForest: Interpreting random forests via visual analytics," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 407–416, Jan. 2019.
- [86] N. Elmqvist and J. Fekete, "Hierarchical aggregation for information visualization: Overview techniques and design guidelines," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 3, pp. 439–454, May/Jun. 2010.
- [87] G. Sun, R. Liang, H. Qu, and Y. Wu, "Embedding spatio-temporal information into maps by route-zooming," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 5, pp. 1506–1519, May 2017.
- [88] A. Wu et al., "AI4VIS: Survey on artificial intelligence approaches for data visualization," *IEEE Trans. Vis. Comput. Graph.*, to be published, doi: [10.1109/TVCG.2021.3099002](https://doi.org/10.1109/TVCG.2021.3099002).
- [89] H. Li, Y. Wang, A. Wu, H. Wei, and H. Qu, "Structure-aware visualization retrieval," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2022, pp. 1–14.
- [90] D. Shi, W. Cui, D. Huang, H. Zhang, and N. Cao, "Reverse-engineering information presentations: Recovering hierarchical grouping from layouts of visual elements," 2022, *arXiv:2201.05194*.
- [91] M. Vartak, S. Huang, T. Siddiqui, S. Madden, and A. Parameswaran, "Towards visualization recommendation systems," *ACM SIGMOD Rec.*, vol. 45, no. 4, pp. 34–39, 2017.
- [92] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-driven documents," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.
- [93] A. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer, "Reactive Vega: A streaming dataflow architecture for declarative interactive visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 659–668, Jan. 2016.
- [94] R. Borgo et al., "Glyph-based Visualization: Foundations design guidelines techniques and applications," in *Proc. EuroGraph. state Art Rep.*, 2013, pp. 39–63.
- [95] D. Li et al., "ECharts: A declarative framework for rapid construction of web-based visualization," *Vis. Inform.*, vol. 2, no. 2, pp. 136–146, 2018.
- [96] G. Li, M. Tian, Q. Xu, M. J. McGuffin, and X. Yuan, "GoTree: A grammar of tree visualizations," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2020, pp. 1–13.
- [97] D. Park, S. M. Drucker, R. Fernandez, and N. Elmqvist, "ATOM: A grammar for unit visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 12, pp. 3032–3043, Dec. 2018.
- [98] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2376–2385, Dec. 2013.
- [99] M. Chen and D. S. Ebert, "An ontological framework for supporting the design and evaluation of visual analytics systems," *Comput. Graph. Forum*, vol. 38, no. 3, pp. 131–144, 2019.
- [100] G. Robertson, D. Ebert, S. Eick, D. Keim, and K. Joy, "Scale and complexity in visual analytics," *Informat. Vis.*, vol. 8, no. 4, pp. 247–253, 2009.
- [101] A. Wu, D. Deng, F. Cheng, Y. Wu, S. Liu, and H. Qu, "In defence of visual analytics systems: Replies to critics," *IEEE Trans. Vis. Comput. Graph.*, to be published, doi: [10.1109/TVCG.2022.3209360](https://doi.org/10.1109/TVCG.2022.3209360).
- [102] R. Amar, J. Eagan, and J. Stasko, "Low-level components of analytic activity in information visualization," in *Proc. IEEE Symp. Informat. Vis.*, 2005, pp. 111–117.
- [103] B. Saket, A. Ender, and C. Demiralp, "Task-based effectiveness of basic visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 7, pp. 2505–2512, Jul. 2019.
- [104] M. Farrugia and A. Quigley, "Effective temporal graph layout: A comparative study of animation versus static display methods," *Informat. Vis.*, vol. 10, no. 1, pp. 47–64, 2011.
- [105] Y. Zhao et al., "Preserving minority structures in graph sampling," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 1698–1708, Feb. 2021.
- [106] C. Collins, G. Penn, and S. Carpendale, "Bubble Sets: Revealing set relations with isocontours over existing visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1009–1016, Nov./Dec. 2009.
- [107] N. Cao, Y. Lin, X. Sun, D. Lazer, S. Liu, and H. Qu, "Whisper: Tracing the spatiotemporal process of information diffusion in real time," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2649–2658, Dec. 2012.
- [108] D. Gotz and M. X. Zhou, "Characterizing users' visual analytic activity for insight provenance," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, 2008, pp. 123–130.

Dazhen Deng received his B.E. degree from Zhejiang University in 2018. He is currently working toward the PhD degree with the State Key Lab of CAD&CG, Zhejiang University. His research interests mainly lie in machine learning for visualization and sports visualizations. For more information, please visit <https://dengdazhen.github.io>.

Weiwei Cui received the BS degree in computer science and technology from Tsinghua University, China and the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology. He is a principal researcher with Microsoft Research Asia, China. His primary research interests lie in visualization, with focuses on text, graph, and social media. For more information, please visit <http://research.microsoft.com/en-us/um/people/weiweicu/>.

Xiyu Meng received the BS degree in thermal engineering from Zhejiang University. He is currently working toward the PhD degree with the State Key Lab of CAD&CG, Zhejiang University.

Mengye Xu received the BE degree in digital media technology from Zhejiang University.

Yu Liao received the BE degree in digital media technology from Zhejiang University. She is currently working toward the graduation degree with the Human-Computer Interaction Institute, Carnegie Mellon University.

Haidong Zhang received the PhD degree in computer science from Peking University, China. He is a principal architect with Microsoft Research Asia. His research interests include visualization and human-computer interaction.

Yingcai Wu received the PhD degree in computer science from the Hong Kong University of Science and Technology. He is a professor with the State Key Lab of CAD&CG, Zhejiang University. His main research interests are in information visualization and visual analytics, with focuses on sports science and urban computing. Prior to his current position, he was a postdoctoral researcher with the University of California, Davis from 2010 to 2012, and a researcher with Microsoft Research Asia from 2012 to 2015. For more information, please visit <http://www.ycwu.org>.

▷ **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.**