Towards Systematic Design Considerations for Visualizing Cross-View Data Relationships

Maoyuan Sun, Akhil Namburi, David Koop, Jian Zhao, Tianyi Li, and Haeyong Chung

Abstract—Due to the scale of data and the complexity of analysis tasks, insight often requires coordinating multiple visualizations (views), with each view displaying different parts of data or the same data from different perspectives. For example, to analyze car sales records, a marketing analyst may use a line chart to visualize the trend of car sales, a scatterplot to inspect the price and horsepower of different cars, and a matrix to compare the transaction amounts in types of deals. To explore related information across multiple views, current visual analysis tools heavily rely on brushing and linking techniques, which may require a significant amount of user effort (e.g., many trial-and-error attempts). There may be other efficient and effective ways of showing cross-view data relationships to support data analysis with multiple views, but currently there are no guidelines to address this design challenge. In this paper, we present systematic design considerations for visualizing cross-view data relationships, which leverages *descriptive aspects* of relationships and *usable visual context* of multi-view visualizations. We discuss pros and cons of different designs for displaying cross-view data relationships, and provide a set of recommendations for helping practitioners make design decisions.

Index Terms—Cross-view data relationship, multiple views, visual analytics.

✦

1 INTRODUCTION

EXPLORING data by coordinating multiple visualizations in different views [1] has been a common approach for data analysis in many fields, such as bioinformatics [2], business intelligence [3], cybersecurity [4], text analytics [5], and time-series analysis [6]. Each visualization supports certain analysis tasks by showing either different parts of data, or displaying the same data from different perspectives. Analysts need to constantly relate information from multiple visualizations to gain a comprehensive understanding of the entire dataset [1].

As an example, suppose an analyst, Emma, is using Jigsaw [5] to explore a collection of intelligence reports. After loading the dataset, Jigsaw provides a *list view* that shows relations between named entities extracted from the documents, a *graph view* that displays connections between the entities and documents, and a *document view* that shows the original text (Figure 1). Emma investigates each view and relates the information from multiple views to explore hidden clues in the dataset. Jigsaw uses coordination-based techniques to support relating data across views. As Emma selects entities in the *list view*, Jigsaw highlights corresponding entities in the *graph view* and the related text in the *document view*. However, there are numerous ways to connect the information from different views. Every time Emma selects a data point, she needs to check highlighted information
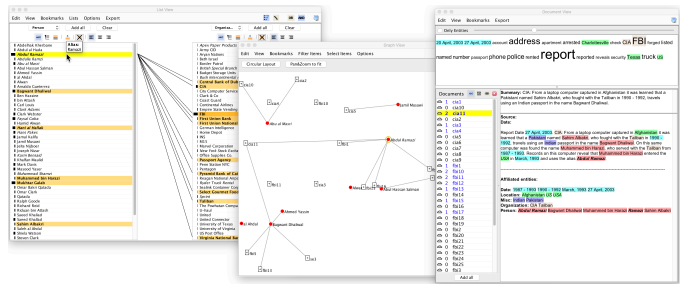

Fig. 1. An example of relating data from three visualizations in Jigsaw: a list view (left), a graph view (middle), and a document view (right).

in other views. After selecting twenty entities, Emma may become confused about which parts of highlighted text in the document view are related to each entity.

Relating data across multiple visualizations is not as straightforward as it looks like. Due to the lack of systematic design guidelines, prior research has primarily relied on brushing and linking based techniques [7], [8], [9], [10], [11], [12], [13]. These techniques are also used in many visual analysis tools (e.g., Caleydo [2], Canopy [14], IN-SPIRE [15], Jigsaw [5], and Tableau [16]). While brushing and linking based techniques can reveal cross-view data relationships, user interactions are limited to visual elements within the views (e.g., specific nodes in a graph) rather than visual relations at the view level. There is a need to support viewing and inspecting visual marks that directly encode cross-view data relationships. Relationship-based visual marks can directly encode data relations across views, reducing user interaction effort. However, visual marks that encode relationships at the view level are usually defined by bundled edges that link multiple data points from different views. This may lead to visual clutter, especially when an increasing number of visual marks are added.

• *Maoyuan Sun, Akhil Namburi, and David Koop are with the Department of Computer Science, Northern Illinois University, DeKalb, IL 60115. E-mail: {smaoyuan, namburi.akhil12, dakoop}@niu.edu.*
• *Jian Zhao is with the School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1. E-mail: jianzhao@uwaterloo.ca.*
• *Tianyi Li is with the Department of Computer Science, Loyola University Chicago, Chicago, IL. E-mail: tli@cs.luc.edu.*
• *Haeyong Chung is with the Department of Computer Science, University of Alabama in Huntsville, Huntsville, AL 35805. E-mail: haeyong.chung@uah.edu.*

This brings a design challenge of visualizing cross-view data relationships. In particular,what important factors do visualization designers and practitioners need to consider to show data relationships across multiple views? To address this, we present systematic design considerations for visualizing cross-view data relationships, which leverage (1) descriptive aspects of relationships and (2) usable visual context of multi-view visualizations. The former formalizes cross-view data relationships with a descriptive framework that highlights four aspects: *schema*, *structure*, *weight*, and *size*. The latter regards usable visual context of multi-view visualizations. Combining both, the design of visualizing cross-view data relationships can be considered as using certain, available visual resource of multi-view visualizations to present some particular aspects of the relationship. Based on this, visualization design options can be systematically examined, which enables comparing different designs.

In summary, this work highlights the following three key contributions:

- We present systematic design considerations for visualizing cross-view data relationships, including both descriptive aspects of relationships and usable visual context of multiple views.
- We analyze a variety of visualization design options, and discuss their pros and cons.
- We suggest a set of recommendations to help visualization designers and practitioners make design decisions.

2 BACKGROUND

We clarify the notion of key terms used in this paper: data relationship, visual element and view. Data relationship is the focus of this work. Visual element and view are major components in multi-view visualizations.

2.1 Data Relationship

Our notion of data relationship begins with the concept of *entity set*. An entity set is a collection of *non-duplicated* data entities that share the same attribute(s) (e.g., a set of images, or a list of people's names). Given two entity sets A and B, a relationship between them, $R(A, B)$, is a subset of $A \times B$, where $\times$ denotes the Cartesian product operation. If $R(A, B)$ is not $\emptyset$ (the empty set), we say that A is related to B. Otherwise, we say that A and B are independent from each other. In different scenarios, the relationship R may be determined differently. For instance, in text analytics, R can be defined as word co-occurrence; while in cybersecurity, R may be determined by communication-based attributes between MAC addresses and web pages. Based on this notion, we identify four key aspects of data relationships (Section 3) and further consider them in the context of multiple views (Section 4). This drives our design considerations as using *what available visual context* in multiple views to show *which aspects* of cross-view data relationships that users care about.

2.2 Visual Element

In a broad sense, a visual element can be considered to be anything that is shown on a display. Based on the visualization reference model [17], visual elements refer to spatial substrates, marks, and graphical properties, which consider possible visual forms mapped from data. Following this rationale, our notion of visual element is *a graphical representation unit that is designed based on data attributes and values*. For example, each node in a scatterplot reveals a data entity and the positions of nodes correspond to values on two specific attributes. Moreover, an aggregation of visual elements can be perceived as major components of a unit visualization (e.g., several spatially organized dots forming a perceived rectangle that indicates a bar in a bar chart) [18], [19]. Thus, visual elements serve critical parts of available visual context of multiple views, where visual encodings and user interactions can be applied to support exploring cross-view data relationships (Section 5.4 - 5.6).

2.3 View

There is no broad consensus on the definition of a view [20]. Based on prior work in information visualization and multiple coordinated views, a visualization view can be understood from the following four perspectives:

- A **process**-centric perspective: *visual mapping product*. A view results from a visual mapping of data [17], [21], and it is considered the final stage of Chi's data state reference model [21] that directly interacts with users.
- A **model**-centric perspective: *data + representation modeling*. A view is a set of data and specifications for displaying them [10], and different specifications lead to different *forms* of visual representations [9], [22].
- A **perception**-centric perspective: *spatial separation*. A view is an independent, separated and bounded area (e.g., windows) [7], [23], where data is displayed in some types of visual forms [20].
- A **task**-centric perspective: *analytics scaffolding*. A view supports users performing certain analytical tasks on data [13], exploring particular attributes of data [24], or recording analytical insights [25].

While these descriptions emphasize different aspects of a view, an important aspect, *semantic coherence*, seems to be missing. We regard semantic as the meaning of a series user actions corresponding to certain tasks [26]. Yet, a view reflects data transformations in some visual forms, and can be shown in a bounded area, intended to separate one from another. Such a boundary becomes vague when considering user tasks. For example, when exploring a scatterplot matrix, it is hard to say whether each scatterplot or the whole matrix is a view. When users try to understand a dataset based on two attributes only, one scatterplot is considered a view. When users attempt to explore eight attributes, it is reasonable to consider the whole matrix as a view.

We define a view as *a collection of visual elements that are spatially organized in a semantically coherent manner to support specific analysis tasks*. A view can be a stand-alone window of a desktop application (e.g., Jigsaw's Graph View [5]), or a bounded area in a web application (e.g., a chart with visible borders in MyBrush [27] and Vega-Lite [28]) that holds a specific visualization (e.g., a scatterplot). Moreover, a view can also be one component of a complex visualization (e.g., a block in Domino [29]) where such a component is perceptually distinguishable and usable for data explorations. Based
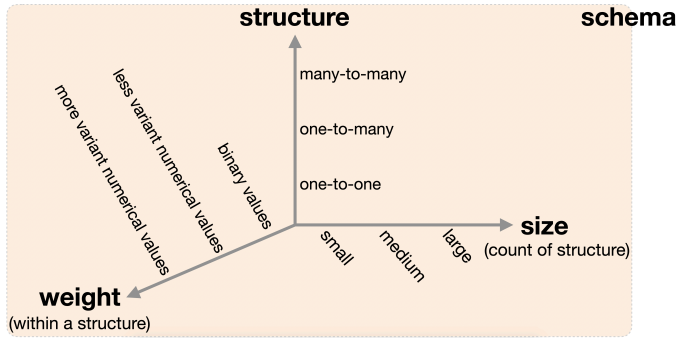
Fig. 2. A descriptive framework of data relationships with four aspects: schema, structure, strength, and size.

on this notion, we have identified three types of cross-view data relationships (Section 3.2) and components of visual context in multiple views (Section 4).

3 DATA RELATIONSHIP FRAMEWORK

In this section, we first introduce aspects that characterize data relationships and then describe different types of cross-view data relationships.

3.1 Data Relationship Characterization

We characterize data relationships as analytical units in a descriptive framework with four aspects (Figure 2): *schema*, *structure*, *weight*, and *size*.

Schema is the overarching strategy used in an analytical task. It considers the *types* of information that are selected from a dataset. For example, a dataset about people and organizations they are related to may include three types of information: person, location, and organization. Data relationships can be constructed between entities of various types of information, such as a relationship between person and organization, or a relationship among location, computer MAC address, and website URL. This is similar to the relational database schema.

The types of information in a relationship-schema are a subset of all possible combinations of different types of information in a dataset. If a dataset has n different types of information (e.g., person, location, organization, date, and phone number), the number of total possible schemata is $C_n^1 + C_n^2 + C_n^3 + ... + C_n^n = 2^n - 2$, where C_n^m denotes the number of ways to combine m types of information from a total of n types of information. Specifically, C_n^1 indicates the relationships of entities of the same type (e.g., clustering persons in an ego-centric network [30]). Thus, the more types of information a dataset has, the greater the number of possible schemata that can be generated from it. Moreover, users may choose to use different schemata for each analysis, so for the same dataset, each user may discover different relationship structures.

Structure is the organization of entities in a data relationship. Different structures can be described by the number of entities in each involved type of information [31], similar to the cardinality of a set.

As is shown in Table 1, there are four types of structures, defined by the number of entities in a data relationship. A

TABLE 1
Four types of relationship structures between visual elements.

Relationship Structure	Relationship Level	Mathematical Format
One-to-One	Individual	$1 : 1$
One-to-Many	*Single*-Group	$1 : e \ (e \geq 2)$
Many-to-Many (simple)	*Bi*-Group	$e : f \ (e, f \geq 2)$
Many-to-Many (complex)	*Multi*-Group	$e : f : ... : z \ (e, f, ..., z \geq 2)$

one-to-one structure is an *individual*-level of relations between two entities. For example, one location in a list view is related to one person in a graph view. It is the most basic structure, based on which higher-level structures can be built. A *one-to-many* structure is a *single-group* level of relationships that associate a group of entities with one individual entity. For example, one location in a list view is related to seven organizations in a graph view. A *many-to-many* structure is a high-level relationship that connects at least two groups of entities. In a simple case at the *bi-group* level, it connects two groups of entities. For example, five locations in a list view are related to seven organizations in a graph view. In a complex case at the *multi-group* level, it relates multiple groups (e.g., [32]). For example, five locations in a list view are related to seven organizations in a graph view, which are mentioned by six documents in a document view.

Weight is the strength of a data relationship. For example, during an investigation, suppose we discover Alan had a conflict with the victim at work, but Ivan owed a huge debt to the victim. We might consider next steps based on the strength of these links between the suspects and victim. Weight measures how strong entities are related. A simple model is binary, where the value is 0 for no relation, and 1 when a relation between two entities exists. In a complex case, the weight can take a continuous value (e.g., in the range $[0, 1]$), where a larger value indicates a stronger relationship between two entities.

Weight can be used to find relationship structures. As discussed by Sun et al. [33], biclusters, a type of many-to-many relationship structure, can be discovered by aggregating entities from two groups, where each entity in one group is related to all entities in the other group (i.e., the values corresponding to all their relations are 1). The weight of a bicluster relation can be measured by the number of entity-wise connections between the two groups. Also, given a relationship structure, we can check its weight by exploring each individual relations within the structure.

Size is the *number* of relationship structures (e.g., ten individual-level relationships, three single-group relationships and five bi-group level relationships). Because we consider relationship structure as a basic analytical unit, size focuses on the number of such units, instead of the count of individual-level relationships within a relationship structure. For example, a person making three calls to a phone number is considered as *one* individual-level relationship.

In summary, the structure, weight, and size of a data relationship are determined by the schema. The complexity of each aspect can impact the visualization design of data relationships.

3.2 Data Relationships across Multiple Views

We define data relationships across multiple views as 1) *between visual elements*, 2) *between views*, and 3) *between visual*
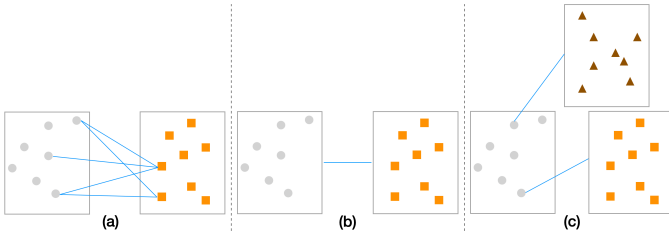
Fig. 3. Three types of cross-view relationships: a) between visual elements, b) between views, and c) between visual element and view. Each box is a view. Gray circles, orange squares, and purple triangles are visual element marks, and blue lines indicate relations.

elements and views (Figure 3).

The first type defines cross-view data relationships as relations between visual elements from different views. Each view has a collection of visual elements, which encode data entities and/or attributes (e.g., nodes in data context map [34]), but there may also be relationships between items in different views represented as relations between the visual elements themselves.

The second type treats each view as an atomic component in a relationship. Relationships between views may reveal high-level insights [35] which go beyond simply aggregating relations between visual elements. For example, in ForceSPIRE [36], each document is shown in a window-based visual metaphor and users can spatially organize documents. The spatialization can reveal certain relations (e.g., documents with similar topics are displayed in a cluster). In this case, the data relationships are defined between different document views rather than between entities within the documents.

The third type involves both visual elements and views. It can be considered as a reflection of Shneiderman's visual information seeking mantra [37]. Consider an overview visualization that shows a scatter plot of the names of people being examined in an analysis. The visual elements in the plot can be related to other views, including tables, maps, and histograms that show detailed information for specific individuals.

While all three types of relationships are important for cross-view analysis, the first type is more challenging to visualize than the other two types. First, it is more granular than the other two types because it involves detailed one-to-one relationship structures between visual elements. The complexity of the different relationship structures can be hard to display and manage. Second, the number of visual elements is often larger than the number of views in real-world cases, presenting scalability challenges. Especially for the three group-level relationship structures between visual elements (rows 2-4 in Table 1), techniques like Euler Diagrams face potential problems [38]. Understanding the design considerations for the first type of relationships across multiple views, which are defined by the relations between visual elements, is therefore essential for visualizing cross-view relationships. In this work, we primarily focus on addressing the above two challenges for visualizing the first type of cross-view data relationship.
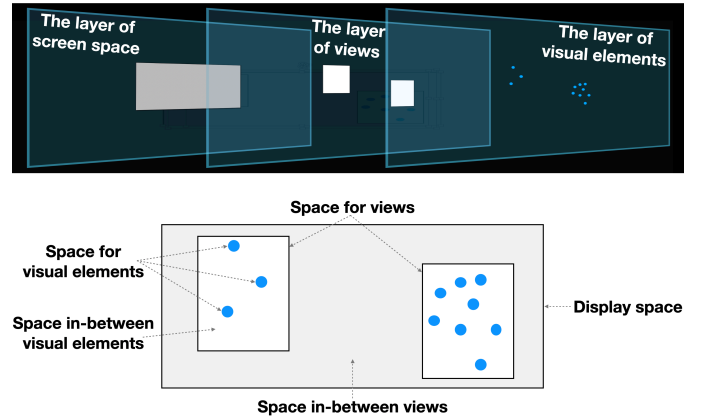

Fig. 4. Three conceptual layers (top) and five perceivable spaces (bottom) in multiple views.

4 VISUAL CONTEXT IN MULTIPLE VIEWS

The previous section discusses characteristics of data relationships, and we next seek visual encodings and user interactions that encode these aspects. Thus, in this section, we explore available *visual context* in multiple views–the locations for possible encodings and interactions.

4.1 Components of Visual Context

We identify three *conceptual layers* with five *perceivable spaces* to characterize the visual context in multiple views.

Conceptual layers refer to spaces along the z-axis in a multiple-view environment (Figure 4 top). We identify three conceptual layers: *screen space*, *views*, and *visual elements*. The *screen space* layer serves a canvas to hold multiple views. The layer of *views* is on top of the *screen space* layer, and is comprised of all views that are shown. Finally, on top of the view layer, there is a layer of *visual elements*; it is home to all visual elements, regardless which views they come from.

Within these conceptual layers, there are five **perceivable spaces** that are areas in xy-space in multiple-view design (Figure 4 bottom). The *display space* allows users to add, remove, and organize visualizations (e.g., the workspace in the analyst's workstation [39]). This space is bounded by the available screen space in a system or tool that supports multiple views. The *space for views* refers to the space taken by the visualizations (e.g., the two white areas in Figure 4 bottom). The *space in-between views* refers to the space between views in the display space. It can be none if the space for views fully cover the display space. Within the space for views, there is the *space for visual elements* and the *space in-between visual elements*. Note that in some cases (e.g., space-filling visualizations), a visualization view can be full of visual elements (e.g., treemaps). In such cases, there is no space in-between visual elements.

4.2 Relations between Visual Context Components

The five perceivable spaces can be mapped to the three conceptual layers. The display space is in the screen space layer. The view and the visual element spaces belong to the layer of views and the layer of visual elements, respectively. The space in-between views is mapped to the screen space

TABLE 2
A summary of our identified design requirements for exploring cross-view data relationships.

Relationship Aspects	Design Requirements
Schema	R1: enabling users to specify a schema to construct cross-view data relationship R2: supporting users modifying an existing schema of cross-view data relationship
Structure	R3: displaying a cross-view data relationship structure R4: supporting users changing a cross-view data relationship structure
Weight	R5: displaying the detailed weight value for each individual-level relation inside a cross-view data relationship structure R6: displaying an overall weight value for a cross-view data relationship structure R7: supporting users changing the weight for each individual-level relation inside a cross-view data relationship structure
Size	R8: displaying many cross-view data relationship structures R9: supporting users managing many cross-view data relationship structures

layer, and is composed of those regions not covered by the projections of view spaces from the layer of views. Similarly, the space in-between visual elements is in the view layer, and is composed of those regions not covered by the projections of visual elements on the top layer. Because the screen space works as a container for views, and each view serves a container for its visual elements, we place the two in-between spaces in these containing layers. The three conceptual layers and five perceivable spaces partition the usable, visual context in multiple-view visualizations. To visualize cross-view data relationships, we can apply visual encodings in the visual context. For example, we can explicitly connect visual elements using visual links (e.g., lines) in the space in-between views. We can further encode properties of the connections by varying appearances (e.g., color) of the visual elements. A clear understanding of the visual context may help visualization designers consider where to show cross-view data relationships.

5   DESIGN FOR CROSS-VIEW DATA RELATIONSHIP

With the *data relationship framework* and the *visual context* of multiple views, we can systematically examine the design of visualizing cross-view data relationships. The data relationship framework characterizes which aspect(s) of the relationship a design may emphasize, and the visual context identifies available resources that a design may rely on. Thus, designing cross-view data relationship visualizations can be considered as *using the available visual context of multi-view visualizations to present particular aspects of the relationship*.

5.1   Design Requirement Analysis

Based on the relationship framework discussed in Section 3, we derived a set of design requirements (Table 2) by considering possible user needs corresponding to each relationship aspect when exploring cross-view data relationships.

For schema, there are two major design requirements: 1) **schema construction** and 2) **schema modification**. The former refers to users needing to specify the schema of cross-view data relations (e.g., a user selects several views to be considered in such relationships). The latter occurs when users make changes to some existing schema. As the usage of different types of visualizations in multiple views is driven by user tasks, users may not always relate information from all displayed visualizations. For example, suppose Emma decides to analyze the relationships between people and locations (*schema construction*) in the intelligence

reports with Jigsaw [5]. She creates two scatterplot views for all the people and locations separately. When she selects a person's name, all locations that appeared in the same document are also highlighted. Soon, she discovers several persons of interest and decides to focus on the relationship between these people (*schema modification*). She disables the synchronization feature between the two scatterplot views. Now when she analyzes the relationship between suspects, nodes in the location scatterplot view are not highlighted.

For structure, key design requirements include: 1) **structure display** and 2) **structure management**. To understand a structure, users need to view its components (e.g., data entities and their detailed relations). Because there are multiple levels of relationships, users need to recognize each of them. For example, suppose Emma noticed that three of the people mentioned in the intelligence reports all had records indicating a visit to the same location (bi-group relationship), and that location is related to three other highly sensitive locations (single-group relationship). In some cases, users may need to revise a relationship structure, e.g. because a computation has an error or the relationship not match the domain knowledge. For example, suppose Emma discovers that one of the suspect uses an alias; she can combine that alias and all the related structures to the current structure.

For weight, users may need to understand two levels: 1) **detailed individual level** and 2) **overall structure level**. The former means that users need to understand the detail of a relationship structure. The latter indicates that users need a summative overview of a structure before digging into its details. For a one-to-one relationship between visual elements (Table 1), the summative overview is the same as the detailed individual level. For higher-level structures, the two-level weights are different. While weight is not always considered when using multiple views (i.e., users may just care about whether two entities are related or not), we argue that it is important to consider the design requirements for weights. First, computed relationships have real-valued weights from probabilities. Designers should have the freedom to choose between an existential (0 or 1) or a more granular description (continuous values) of the data relationships. Second, the overall structure level weight serves as a summative overview of multiple individual-level relationships. Thus, it is related to the design requirements of the other components in the data relationship framework. Even at the detailed individual level, users often need to **change weight**. For example, a relationship between two people might become more important as the analysis unveil more hidden clues. This can also further influence

the weights at the overall structure level. Moreover, the computed weight is not always aligned with user domain knowledge. Visualizing such disparities is thus important for discovering new knowledge and making progress in the analysis.

For size, the key design requirement is to support users in seeing, exploring, and navigating many cross-view data relationships, instead of each individual relationship. For real-world analysis problems, it is rarely the case that only one relationship structure exists, so users need to *see and manage many relationships* for analysis. To fulfill this goal, users need to first organize the data relationships by the type of relationship structures. Then, users may want to manage them based on their understanding. For example, after analyzing several person-location relationships and person-person relationships, Emma discovers that these relationships all point to a coordinated crime. She aggregates these two types of relationship structures and proceeds with further analysis.

In summary, we have identified 9 design requirements (Table 2). They offer a systematic and organized view of critical user needs when exploring cross-view data relationships. It is important to note that these design requirements are not an exhaustive list of design requirements. It is also possible that one design does not cover all the requirements as different analyses may focus on different aspects of cross-view data relationships. Following these requirements, we next investigate the design patterns for visualizing cross-view data relations.

5.2 Method of Collecting Related Designs

To examine designs for visualizing cross-view data relations, we recursively searched papers in related fields including information visualization, visual analytics, and human-computer interaction. We defined a set of criteria to determine the relevance of a paper. Based on this, we collected a small group of closely related papers as seed papers, checked their references and papers that refer to them, and then used our criteria to find more relevant papers. We performed this recursively with a maximum recursion depth of three. In case this search strategy favored relatively, highly cited papers, we also searched Google Scholar using the keywords, "coordinated multiple views", "brushing and linking", "small multiples", "dashboard visualization", and "set visualization" (we consider set visualizations as group-level relations that may overlap by sharing visual elements). For each keyword, we checked the results from the first 10 pages, used our criteria to identify related papers, and performed further recursive searches.

Our criteria for determining the relevance of a paper are:

- Visualizations should focus on a 2D layout, instead of 3D or in an immersive environment.
- The work should include a software prototype that uses multiple views and clearly describes its design and interactive features. As discussed in Section 2.3, we consider view broadly. This ensures that a wide range of papers are initially included.
- The work should have key contributions to the design and usage of multi-view visualizations. This helps us identify a small and focused group of papers as seeds for further searches.

- The work should have an in-depth discussion about relating information from multiple visualizations. This helps us exclude work that is simply using these multiple views, brushing and linking, or small multiples, since these are common in many visual analysis tools. We consider a paper having an in-depth discussion, if it covers two or more of the following.
  - Specification of cross-view data relationships, including types of relationships and relationship structure.
  - New designs for exploring or relating data from multiple visualization views.
  - Evaluation of designs or techniques for exploring or relating data from multiple visualization views.
  - Lessons learned in the process of design and development of visual analysis systems in which relating data across multiple views is a key design concern.

Our search started with eight closely-related seed papers, including Improvise [40], Snap-Together [41], Cross-filtered Views [42], VisLink [43], Visual links across applications [44], ConnectedCharts [24], MyBrush [27]. and Domino [29], and ended with 32 relevant papers in total. Improvise and Snap-Together highlight critical contributions to the design of multiple coordinated views, and Cross-filtered Views is a typical example of using brush and linking techniques for filtering data across views. VisLink is a representative example that explicitly links visual elements across different views, while Visual links across applications and MyBrush offer more flexibility in direct, visual linking by including more types of visualizations and allowing users to control the types of the links, respectively. ConnectedCharts presents a design space where multiple bar-based charts are linked and identifies two types of connections, so it is closely related to this work. In addition, Domino supports users interactively and progressively building connected multiform visualizations, which covers different levels of relationships by using a number of visual encodings besides simple lines.

Although this not an exhaustive list, it includes a variety of related work that supports our examination of possible designs for displaying cross-view data relationships. Figure 5 summarizes them. Each row corresponds to a paper, and columns are categorized into three groups. We manually adjusted the order of rows to place examples with similar designs near each other.

5.3 Design for Schema Construction and Modification

We identified four designs that support users specifying the schema of cross-view data relationship.

Automatically including all displayed visualizations is the simplest way for users to manage cross-view data relationship schema. Users do not need any prior knowledge about data under investigation, and they can perform flexible explorations by trying the different visualizations that an analysis tool supports. As users open or close a visualization, it is added to or removed from a schema automatically. Moreover, when users load a pre-defined set of visualizations (e.g., semantic substrates [45]), this design allows the analysis tool to automatically generate a fixed schema that include the whole set of visualizations.

● indicates an example has a visual encoding that uses a visual context and can meet a design requirement

○ indicates an example has a visual encoding that uses a visual context and *needs interactive effort* to meet a design requirement

Figure 5 — table legend and matrix:

Group	CROSS-VIEW DATA RELATIONSHIP ENCODING					DESIGN REQUIREMENTS								VISUAL CONTEXT USAGE		
Sub-group	ADDING RELATIONSHIP MARKS		UPDATING VISUAL ELEMENT CHANNEL		ENRICHING VISUAL ELEMENT MARK	SCHEMA		STRUCTURE		WEIGHT		SIZE		VISUAL ELEMENT SPACE LAYER	VIEW SPACE LAYER	SCREEN SPACE LAYER
Column	INDIVIDUAL-LEVEL RELATION MARK	GROUP-LEVEL RELATION MARK	COLOR	POSITION		R1: SPECIFY A SCHEMA	R2: MODIFY A SCHEMA	R3: SHOW A STRUCTURE	R4: MODIFY A STRUCTURE	R5/R6: SHOW WEIGHT OF A RELATIONSHIP	R7: MODIFY WEIGHT OF INDIVIDUAL-LEVEL RELATION	R8: SHOW MANY STRUCTURES	R4: MANAGE MANY STRUCTURES	SPACE FOR VISUAL ELEMENTS	SPACE IN-BETWEEN VISUAL ELEMENTS	SPACE IN-BETWEEN VISUALIZATION VIEWS

Rows (examples):

- VISLINK [43]
- SHOW ME THE INVISIBLE [56]
- CONTEXT-PRESERVING VISUAL LINKS [52]
- COLLABORATIVE INFORMATION LINKING [55]
- VISUAL LINKS ACROSS APPLICATIONS [44]
- CONNECTEDCHARTS [24]
- MYBRUSH [27]
- CALEYDO [2]
- SEMANTIC SUBSTRATES [45]
- GPLOM [64]
- FLOWSTRATES [65]
- SAVIL [53]
- JIGSAW [5]
- STRATOMEX [57]
- DOMINO [29]
- VISBRICKS [51]
- MATCHMAKER [50]
- FURBY [58]
- BISET [33]
- CNNVIS [54]
- BIDOT [67]
- BIXPLORER [46]
- NODETRIX [69]
- MAPTRIX [66]
- GRAPHTRAIL [25]
- PANORAMICDATA [76]
- IMPROVISE [40]
- SNAP-TOGETHER [41]
- CROSS-FILTERED VIEWS [42]
- AGGRESET [77]
- KESHIF [78]
- VISTILES [73]

Fig. 5. A summary of our identified designs based on the collected papers.

Incrementally expanding from a visualization supports users gradually building a schema. With a visualization currently under investigation, this design assumes that users want to follow the lead of identified, useful information in the current visualization, and perform further exploration. It allows users to request more relevant information by interacting with the visualization, and an analysis tool automatically finds and displays this in another visualization (e.g., Bixplorer [46], ConnectedCharts [24], Domino [29], and GraphTrail [25]). In such an exploratory process, a cross-view data relationship schema is incrementally specified. If users consider a newly added visualization useless and close it, it is removed from the schema.

Building while investigating a few visualizations provides rich visual guidance for users to specify and modify cross-view data relationship schema. Different from incremental expanding, this design allows users to see multiple visualizations, and then decide which of them to include in a schema. Compared to expanding from one visualization, multiple visualizations may offer more guidance for users to make decisions. To support user decisions, each visualization has a widget (e.g., a checkbox or a switch button) to control its inclusion.

Creating a schema before showing any visualizations is a design that asks users to specify a schema first. Then, related visualizations are displayed accordingly. Compared to other designs, it requires users to have some knowledge about the dataset being explored. It has been implemented as a configuration panel in an analysis tool (e.g., the snap specification dialog in Snap-Together [41]). Here, users use the panel to modify a schema instead of directly interacting with visualization views.

For schema modification, we have identified two major designs. One relies on displayed views, while the other is independent from views. A *view-dependent* design requires users interacting with a view to determine whether a view is included in a schema or not (e.g., opening or closing a view, moving one view close to or away from another, or turning on or off of coordinating a view). A *view-independent* design allows users using a configuration panel to modify a schema. A view-dependent design supports users modifying a schema while investigating a view, but the schema is not explicitly revealed. The view-independenent design does show the schema but requires users to switch between visualizations and the configuration panel which may interfere with explorations.

5.4 Design for Structure Display and Management

We have identified three key design concepts to show cross-view data relationship structure: (1) *adding relationship*
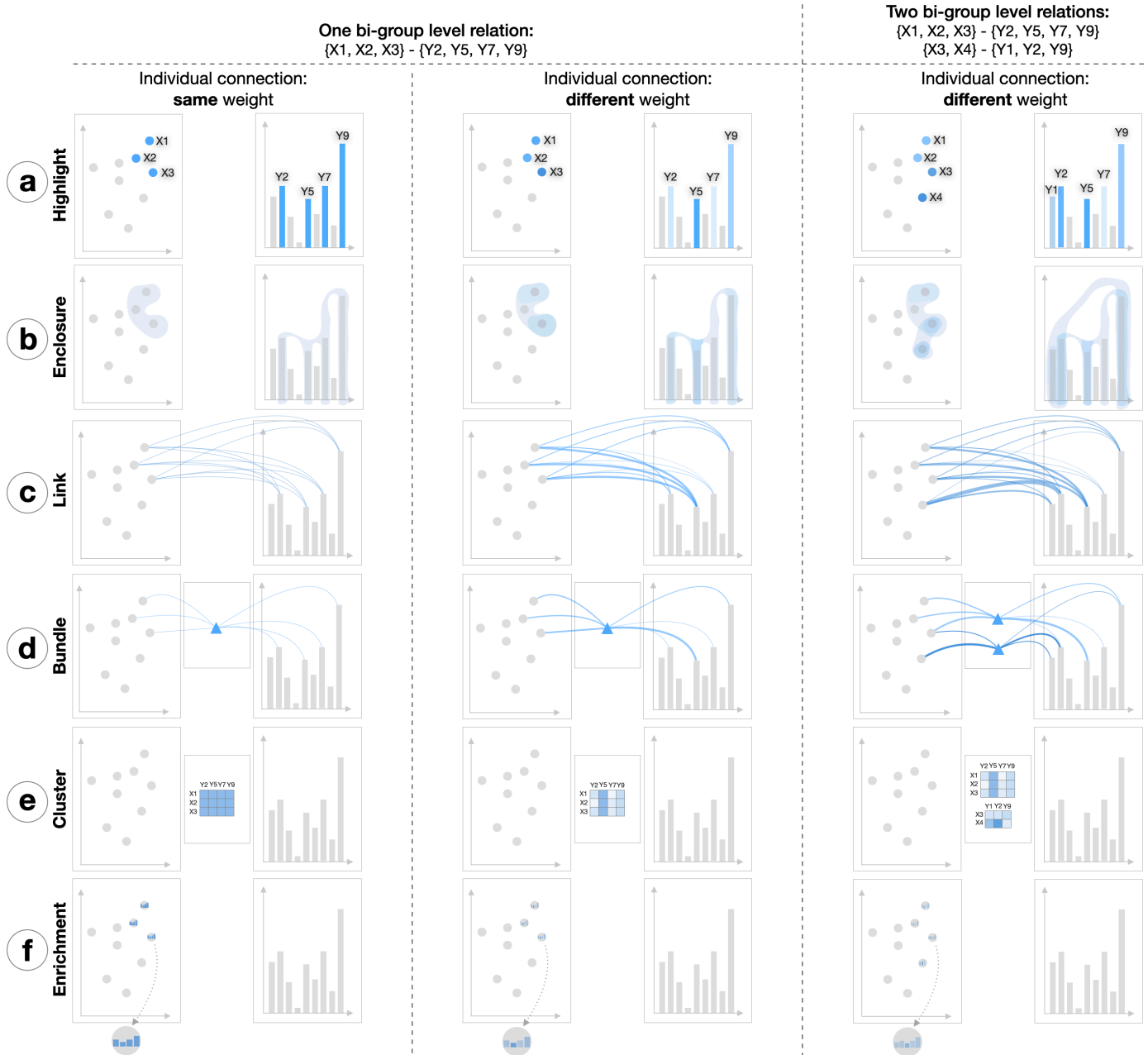
Fig. 6. Examples of visual encodings for cross-view data relationships: (a) highlight, (b) enclosure, (c) link, (d) bundle, (e) cluster and (f) enrichment.

marks, (2) *updating channels of existing visual elements*, and (3) *enriching marks of existing visual elements*. (In this paper, *mark* and *channel* follow Munzner's definitions [20].) The design concepts emphasize different visual encodings for relationship structure and need not be mutually exclusive, instead working together. Over 80% of our collected references used more than one of the design concepts. Figure 6 provides an overview of these design concepts with six concrete examples. (b)–(e) reflect design concept (1) by adding different types of marks that encode cross-view relationships. (a) corresponds to design concept (2) by using highlighting. (f) shows design concept (3) by enriching the marks for visual elements in a view. These examples are not exhaustive; other visualizations may use other visual encodings.

5.4.1 Adding Relationship Mark

Adding relationship marks is the most straightforward design concept, as the marks directly encode cross-view data relationship structures. There are two types of relationship marks: *individual-level* marks and *group-level* marks. Both may use the space in-between visual elements and the space in-between views. Figure 7 shows examples of possible designs for relationship marks, organized by rows. For each row, different columns present design alternatives.

An **individual-level relationship mark** is a line, which visually links visual elements from two views. As is shown in the top row of Figure 7, such a line can have multiple forms: straight, stepwise, curved, or "broken" (as people can still perceive continuity based on the Gestalt principles [47]).
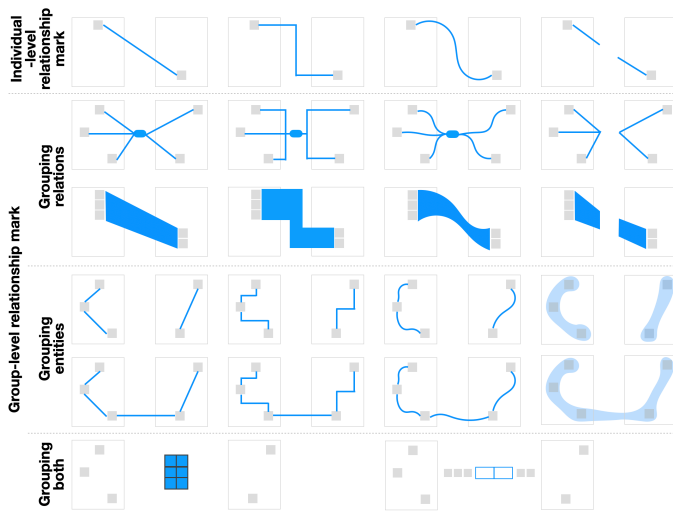
Fig. 7. Examples of possible designs for relationship marks.

Compared with straight lines, curved lines may improve aesthetics [48] and stepwise lines with orthogonal edge routing may improve chart readability [49], although this uses more pixels of the display space. While "broken" lines may produce fewer edge crossings than other forms, they may require more user effort to identify connected visual elements due to the gaps.

An individual-level relationship mark encodes an individual-level relationship structure. It can also be used to link the same set of entities across multiple views. Multiple individual-level marks can help users perceive bi-group level relationship structure (Figure 6(c) left), but large numbers of marks may cause visual clutter and line crossings. When more than one bi-group level relationship structures exists, using a collection of lines cannot effectively reveal different structures. For example, in Figure 6(c) right, users have to trace lines and check shared elements to identify two different bi-group level structures. To better serve complex relationships, group-level relationship marks have been created and studied [50], [51], [52], [53].

The design of group-level relationship marks include three grouping strategies: *grouping relationships* (Figure 6(d) and Figure 7 row 2 and 3), *grouping entities* (Figure 6(b), and Figure 7 row 4 and 5), and *grouping both entities and relationships as a cluster* (Figure 6(e), and Figure 7 bottom row). These strategies can reveal bi-group level relationship structure, but their relative effectiveness depends on properties of the visualization including the layout of visual elements and the availability of the space in-between views.

A **relationship-grouping** design leverages individual-level relationship marks. It emphasizes grouping individual relations. Layouts of visual elements in views impact the design of relationship groupings. Based on whether visual elements are neighboring and aligned in the 2D space, there are two different designs: **edge bundles** and **ribbons**. Edge bundles (Figure 7 row 2) use the the space in-between views [2], [33], [54] or the space in-between visual elements [55], [56]. Ribbons link two groups of visual elements; within each group, visual elements are neighboring and aligned (Figure 7 row 3). Because grouping is applied to sets

of individual-level relationships, the display of individual-level relationship marks impacts the design of group-level relationship marks. Thus, there exist design variations, including different edge bundle and ribbon representations (e.g., straight [57], curved [50], [51] and "broken" [58]). Compared with edge bundling, ribbons reduce the number of displayed lines but are not as flexible as edge bundling due to layout requirements.

Similar to set visualization techniques [38], an **entity-grouping** based design uses marks to group visual elements of a bi-group level relationship. These marks can be lines [59] or areas [60], [61], [62], [63], [64] that link or enclose visual elements of the same relationship. For a bi-group level relationship, if we consider the groups of related visual elements as multiple sets, the marks group visual elements in the same view (Figure 7 row 4). If we consider them as one set, the marks group them all (Figure 7 row 5). Compared to edge bundling, grouping entities reduces the number of marks to be added as shown by the difference between row 2 and rows 4 and 5 in Figure 7). However, this grouping cannot show detailed cross-view connections between visual elements.

The third strategy, **grouping both relationships and entities**, emphasizes creating marks that aggregate and reveal both aspects of a relationship structure. Due to this aggregation, such marks reside in the space in-between visualization views. Moreover, unlike the first two strategies, marks created with this strategy may duplicate related visual elements. Possible designs for this strategy include MapTrix [65], [66] and BiDot [67] where related entities are connected via lines through a matrix.

Adding relationship marks provides three key benefits. First, this design enables users to see relationships through added marks that are distinct from existing visual element marks. Second, it can be adapted to different levels of relationships and include encodings to reveal the weight of a structure. Third, the added relationship marks can serve as interaction handles that support users directly manipulating cross-view data relationships (e.g., by dragging edge bundles [68] or matrices [69] and merging similar bundles [70]). However, adding relationship marks has two major drawbacks. First, it may cause visual clutter from the added relationship marks (e.g., many edge crossings). Second, newly added relationship marks may impact user perception of *existing* visual elements in multiple views, especially those that overlap existing visual elements.

5.4.2 Updating Channels of Visual Elements

In contrast to adding relationship marks, updating channels of visual elements uses *existing* visual elements that already exist in the views. This concept involves updating certain channels of visual elements to attract user attention to relationships (Figure 6 (a)). Two channels are commonly used: **color** and **position** (primarily in the space for visual elements).

When using color, visual elements of cross-view relationships are highlighted [71], [72]. Color hue can be used to differentiate multiple relationship structures. The highlight-oriented design preserves the layout of visual elements, and it helps users distinguish between those visual elements in cross-view relationships are those that are not. Other
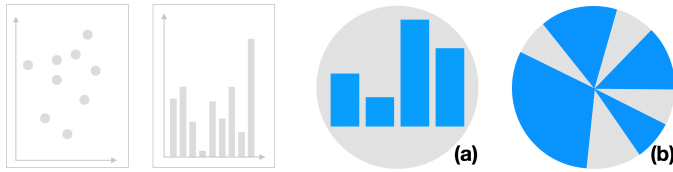
Fig. 8. Design examples of enriching visual element marks. (a) shows the design of filling a visual element mark in a scatterplot with related visual element marks from a histogram. (b) demonstrates the design of transforming a visual element mark in a scatterplot into a pie chart.

design possibilities to relationships visually salient include changing the size, shape, or texture of visual elements. In addition, these highlighting strategies can also be used to link the same set of entities in multiple views.

When using position, related visual elements are placed near each other. This can also include changing a user's focus area or shifting the positions of related visual elements in one view, as users interact with another view. Examples of changing the user focus area include synchronized scrolling (e.g., Snap-Together [41] and VisTiles [73]) and focus+context in multiple views [74]. A typical example of shifting positions of related elements is Jigsaw's List View which can automatically move related entities to the top of a list as a user selects an entity in another list.

This design concept has two advantages: it avoids visual clutter caused by newly added marks, and it preserves existing visualization designs. However, it suffers from three issues. First, it is hard for users to see detailed cross-view connections between visual elements, so they only get a rough idea of cross-view relationships. Second, it is hard for users to understand the weight of a relationship structure. Third, it is difficult for users to recognize different relationship structures. For example, in Figure 6 (a) right, users not provided guidance are likely to see a single relationship despite the existence of *two* bi-group level cross-view relationships.

5.4.3   Enriching Marks of Visual Elements

Enriching visual element marks is a design concept where other marks (as guest marks) are placed on top of existing visual element marks (as host marks). This concept only uses the space for visual elements in contrast to adding relationship marks with the entity-grouping strategy (since relationship marks use the space in-between visual elements). There are two major types of enrichment: **filled with related visual element marks from another view**, and **transformed into a meaningful mini chart**. The former reveals connections between a host mark and one or multiple guest marks with a containment spatial organization. The latter shows in-depth detail of the data a host mark encodes. Figure 8 shows examples of both.

Like the design of nested views in composite visualizations [75], guest marks in one view can be the same as related visual element marks in another view (Figure 8 (a)). This directly reveals an individual-level relationship structure (between one guest mark and one host mark), or a single-group level relationship (between multiple guest marks and one host mark). Moreover, guest marks can also be made by considering data that a host mark encodes [76],

[77], [78]. In this case, based on cross-view data relations, it provides more detail of the host, revealing it in a mini chart (Figure 8 (b)), rather than simply "borrowing" visual element marks from another view.

Compared to adding relationship marks, the mark-enrichment design avoids visual clutter caused by crossing lines or ribbons. It can better reveal details of a relationship structure than updating channels of visual elements. When using guest marks, it is also possible to use color saturation or luminance to encode weight (Figure 6 (f) middle). In addition, as guest marks are placed on top of host marks, layouts of existing visual elements from related views are well preserved. Finally, since newly added marks are placed *inside* visual element marks of existing visualizations, it reduces the user effort involved in context switching when exploring multiple views [10].

This design concept has three limitations. First, not all marks can be easily enriched, especially those taking up a small area, since guest marks need enough space to be visible. Second, while the containment-based spatial organization works for individual-level and single-group level relationship, it is difficult for users to recognize different bi-group level relationships. Third, newly added marks for enrichment may impact user perception of existing visualizations, especially when a large number of visual elements are enriched.

These design concepts lead to a variety of design options for encoding cross-view data relationship structures. Table 3 summarizes their pros and cons. Because they can work together, multiple choices from different design concepts can be applied together (e.g., using connecting related visual elements with link marks and highlighting them).

5.5   Visual Encodings for Relationship Weight

Additional visual encodings can be used to reveal relationship weights in visualizations designed using the three structure display concepts. The weights of a relationship can be quantitative or ordinal (e.g., strong, medium and weak), and in either case, it is reasonable to use magnitude channels [20].

We consider five magnitude channels that can be applied on top of the encodings for cross-view relationship structures: **color saturation**, **color luminance**, **1D size**, **position on unaligned scale** and **position on common scale**. Based on perception studies, position on common scale is the most effective and color saturation and luminance are the least effective [79]. Thus, position on common scale work best but other encodings like color saturation/luminance may be fine especially if detailed weight judgments are not required. Figure 9 shows design examples that use one of these channels based on the visual encodings of relationship structure discussed in Section 5.4.

When relationship structures are encoded as newly added relationship marks, four of the five channels, exlcuding position on unaligned scale, can be used to show weight. Specifically, we can use thickness (1D size) or color saturation/luminance of a line to encode individual-level relationship weight (Figure 9 (c) and (d)), and color saturation/luminance of ribbons, area marks, and matrix cells to display weight of a bi-group level relationship (Figure 9

TABLE 3
A summary of pros and cons of the three design concepts for visual encodings of cross-view data relationships.

|	Pros	Cons
Adding Relationship Marks	- Users can distinguish visual elements from their relationships with separate marks. - Detailed connections between visual elements and weight are clearly displayed. - Users can directly manipulate relationship marks.	- Added relationship marks can lead to visual clutter. - Added relationship marks may impact user perception of existing visualizations.
Updating Visual Element Channels	- There is no visual clutter from additional marks. - Existing visualizations are well preserved.	- Users cannot see detailed cross-view connections between visual elements. - Users may have difficulty checking the weight of a relationship structure. - Users may have trouble visually separating different relationship structures.
Enriching Visual Element Marks	- Users are directly shown individual-level and single-group level relationships. - Users can see in-depth detail of enriched visual elements. - The layout of visual elements in existing visualizations is well preserved. - Users can better deal with context switching when exploring multiple views.	- Not all visual marks can be easily enriched, especially those with a small area. - Users may have difficulty discriminating between multiple bi-group level relationship structures. - Newly added marks may impact user perception of existing visualizations.
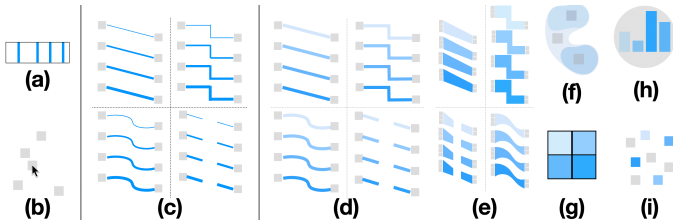
Fig. 9. Examples of visual encodings for relationship weight. (a) uses of position on common scale. (b) uses position on unaligned scale channel. (c) encodes weight using line thickness (1D size). (d)–(i) uses color saturation.

(e), (f) and (g)). In addition, we can use the design of BiDot [67] to precisely show the weight of a bi-group level relationship. It uses a box as a common scale, which holds a set of vertical lines. Each line represents an individual-level relationship, and its horizontal position in the box encodes weight (Figure 9 (a)).

When relationship structures are displaying using the design of updating channels of visual elements, three channels (position on unaligned scale, color saturation, and color luminance) can be used to communicate weight. Relationship weight can be presented as the relative distance between visual elements [36], [39], [80] (Figure 9 (b)). Also, differences in color saturation/luminance of visual elements can indicate different relationship weights (Figure 9 (i)). Similarly, when using the design of enriching visual element marks, the color saturation/luminance of guest marks can be used to present weight.

5.6 Interactions for Handling Many Relationships

To help users recognize and manage relationship structures, user interactions can compliment the visual encodings of cross-view data relationship structures. For the design of adding relationship mark, newly added marks can serve as handlers for users to directly manipulate relationships. Possible interactions include: selecting or hovering over a relationship mark to check its related visual elements, selecting multiple relationship marks for filtering and comparison, moving relationship marks to spatially organize them, dragging one relationship mark on top of another to merge them, and splitting a relationship mark.

For the other two design concepts, which do not add new relationship marks, enabling user interactions on visual elements helps to address the problem that multiple relationship structures cannot be clearly revealed. Brushing and linking is a typical example here; when users select a visual element, related elements are highlighted. If users are allowed to make multi-selections, the highlights can

accumulate so users can recognize different structures by iteratively selecting visual elements and checking those highlighted. At the same time, brushing and linking avoids visual clutter issues caused by added relationship marks and preserves existing views. Moreover, brushing and linking supports highlighting the same set of entities in multiple views (e.g., exploring nodes in a scatterplot matrix) based on a one-to-one mapping.

6 DESIGN RECOMMENDATIONS

Based on the analysis of possible designs for cross-view data relationships, we provide a set of design recommendations. In summary, we suggest considering: 1) structure, weight and size of cross-view data relations, 2) complementary designs, 3) available in-between space, 4) an overview of cross-view data relationships, and 5) existing visualization views. The first two concern the descriptive aspects of relationships, while others concern usable visual context of multiple views.

6.1 Making Data-Driven Design Decisions

It is important to make data-driven design decisions. Given the framework discussed in Section 3, structure, weight and size are the three aspects that impact design decisions. Essential factors related to these aspects include: level of relationship structures, variance of weight values, and the number of relationship structures to be displayed.

Performing a pre-design analysis of cross-view data relationships to gain an in-depth understanding of these factors is helpful to make reasonable design decisions. It is possible that not all levels of structures are present in a given dataset, and not all of them may be useful for user analysis tasks. Thus, a clear understanding of important levels of relationship structures can help narrow the design options down to a focused group.

It is necessary to consider possible values for weight (e.g., whether it is binary or quantitative) and the variance of these values. For example, if bi-group level relationships are most important and weight is binary, we consider designs of adding group-level relationship marks. Because this variance is low, we can focus on the design of grouping relations (e.g., bundling lines). In contrast, if the weights have high variance, lines are likely a poor option and we need to consider other designs like matrices.

The size of cross-view data relationships also impacts design decisions. As shown in Figure 6, when there is a single relationship, all designs clearly reveal it. However, for multiple relationships, not all designs enable users to effectively recognize different structures. When a user needs

to explore a large number of cross-view data relationships (common for real-world analysis problems), adding relationship marks offers better design choices than the other two design concepts, .

6.2 Considering Complementary Design Options

Choose design options that can be complementary to each other, instead of relying on one specific design, especially when exploring more than one relationship aspect. Using multiple design options can better reveal desired aspects of cross-view data relationships (e.g., adding individual-level relationship marks and using color saturation on these marks to reveal relationship structure and weight). It is difficult to find a single design that can meet all design requirements. As discussed before, each design option has its own advantages and drawbacks to reveal the structure, weight, and size of cross-view data relationships. However, they can work together and be complementary to one another. In our analysis of collected, existing designs, a majority made combining multiple options in their work.

To identify complementary designs, a clear understanding of analysis tasks is critical (e.g., which aspects of cross-view data relationships users care about). For example, if users need to check a large number of bi-group level relationships and compare their details, we can choose designs from adding relationship marks and updating visual element channels, and apply them together. Here, we can use matrices to show relations, encoding weight as the color saturation of cells and enabling user manipulation to make comparisons between relations.

6.3 Considering Trade-offs of the In-Between Space

We need to consider two trade-offs in using the in-between space (the space in between views or visual elements). First, the availability of in-between space dictates how well different designs work. If the layout is fixed and views abut one another, there is no in-between space. For adjustable layouts, we can better manage the in-between space, but this often requires layout management support, and it may be unnecessary in some cases. User-analysis tasks help guide whether a design may benefit from different layout strategies.

Second, even when in-between space is available, it may be better to leave it empty instead of using for cross-view linking. The advantage in adding marks is that users can directly see relationships, but it also has drawbacks. First, adding cross-view links may cause visual clutter (e.g., too many lines and line crossings). Second, displaying cross-view links in the in-between space may interfere with user perception of the existing multiple views. White space helps segment different parts of the visualization and can improve understanding, but limits the direct encoding of links between views.

6.4 Creating An Overview

If possible, create an overview of cross-view data relationships, since it can guide user exploration [37]. With a clear understanding of relationship levels that users care about, it is possible to compute them and further display the results
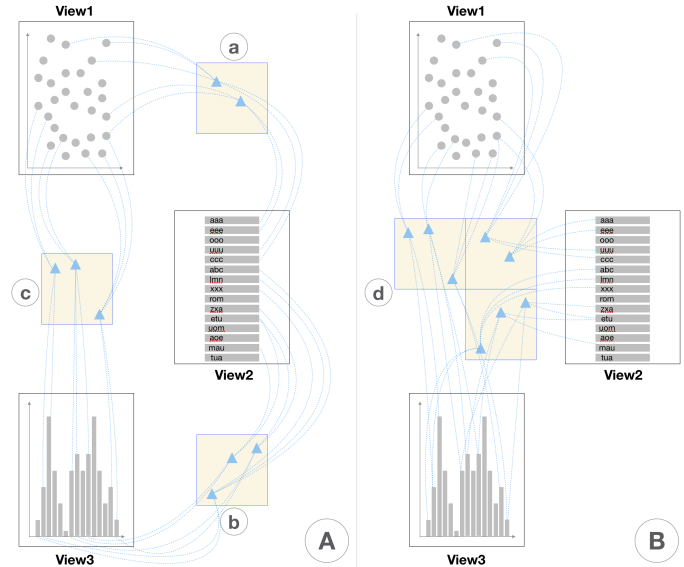

Fig. 10. Two possible designs for overviews of cross-view data relationships: multiple, separated stand-alone overviews (A) and a centralized stand-alone overview (B). Note: (a)-(d) present stand-alone overviews where visual elements indicate cross-view data relationships, and blue lines reveals connections between relationships with their members.

as an overview. Since the design concepts of updating visual element channels and enriching visual element marks rely on displayed visual elements, showing an overview on top of these views is not a good choice. First, as discussed before, it is hard for users to distinguish different relationships with them. Second, it confuses users because an overview of cross-view data relations is shown as a collection of existing visual elements.

However, the design of adding relationship marks in the in-between space supports creating an overview of cross-view data relationships. Using available in-between space, a stand-alone view can be created to hold newly added relationship marks together. Such a view provides an overview of cross-view data relationships. As computed relationships and existing visual elements are separated into different views, existing views are well-preserved and user perception of them are not interfered with by newly added marks. For example, such an overview helps to address Emma's problem while exploring cross-view data relations discussed in Section 1. With more visual guidance, she does not need to manually track highlighted visual elements in different views.

There are two possible designs of such overviews (Figure 10). When there are more than two views, we can create either multiple pairwise stand-alone overviews, or one centralized stand-alone overview for all views. Relationship marks in such overviews can be linked to related visual elements via lines and/or interactive highlighting. Moreover, it is possible to compute the layout of relationship marks and encode their similarity with relative distance which supports users exploring similar relationships.

6.5 Considering Visualizations Used in Multiple Views

Make design decisions by considering visualizations used in multiple views based on two aspects: 1) whether they

are the same type of visualizations and 2) how they and their visual elements are spatially organized. These aspects impact design choices. For example, as shown in Figure 7, when visual elements are spatially aligned, it is possible to replace relationship marks from a collection of lines with one ribbon. However, when visual elements are organized with a graph-based layout and do not neighbor each other, we need to avoid using ribbons.

In addition, the way that multiple views are spatially organized, both in terms of in-between space availability and layout malleability. For example, if the layout of multiple views is fixed and they are juxtaposed with each other (e.g., MyBrush [27]), there is no space in-between visualization views. Thus, we should not try to design the stand-alone overview mentioned before. However, if the layout of multiple views can be flexibly changed, it is possible to modify the space in-between views (e.g., by dragging and moving views), which leads to more design considerations.

When visualizations used in multiple views are of the same type (e.g., a collection of matrices and multiple lists of entities), we may be able to consider specific interaction techniques besides those discussed in Section 5.4 to support users exploring cross-view data relations. For example, it is possible to allow users to drag and move one matrix on top of another, generating an overlay that enables users to see differences between them [81], which is a common task when exploring group-level relationships. However, if visualizations are of different types, this overlay design likely will not work (e.g., users can get confused by overlaying a line chart on top of a graph).

7 DISCUSSION

7.1 Compared to Designs for Composite Visualizations

A primary goal of our design considerations is to help visualization researchers and practitioners leverage and display cross-view data relationships of items across multiple visualization views for supporting data analysis tasks. Earlier, Javed et al. [75] presented a design space called composite visualization views (CVVs), which concerns visually representing data relationships among the same or different visualizations within one visualization view. Both CVVs and our design considerations address the design challenge of visualizing data relationships among multiple views. Specifically, both design approaches present a set of designs, which are coupled with available visual representations and depict the correlation between visual elements in two or more visualizations.

However, there are key differences between CVVs and our design considerations. CVVs focus on the design for combining or coordinating multiple visualizations within a single visualization view. While CVVs cover multiple view visualizations, they deal with multi-view visualizations in which the layout of multiple views is fixed and coordinated (e.g., [27], [45] and [82]). In contrast, we consider the availability of empty space (called the "in-between space") among views or visual elements and the possibility of preserving the context of individual visualization views, with cross-view visual representations. In our design considerations, it is possible to change the layout of visualization views (e.g., by dragging and moving visualization views),

instead of a fixed layout. These specific characteristics of our design considerations offer potential paths to further improve visual analysis applications.

7.2 Towards the Design for Using Multiple Displays

The in-between space and draggable visualization views, if available, enable users to flexibly arrange the positions of views, potentially impacting design decisions. This aspect of our design considerations facilitates a consideration of expanded space by using physical space or large displays in visual analytics tools. Our design considerations suggest different ways for users to freely arrange views into certain layouts based on their relationships (e.g., using alignment, clustering, or ordering). In so doing, the spatial organization of visualization views facilitates the development of more cohesive insights, based on the spatial positions of a group of independent views on the screen [39].

The in-between space serves a useful resource to holds visual encodings for data relationships among visualization views, even when using multiple displays. From our analysis of existing designs, we note that the use of in-between space allows newly added relationship marks (e.g., lines, ribbons, and edge bundles). These relationship marks can potentially provide another conceptual layer of data relationships, thereby further extending the context of multiple views on multiple displays. Particularly, this aspect of our design considerations can address challenges associated with visual analysis using multiple displays. In a multi-display environment (MDE), information and tasks may often be scattered and disconnected among physically separated displays. Thus, an inherent design challenge associated with visual analysis is relating and synthesizing information across separate displays [53]. In this regard, the relationship marks present various strategies for connecting information and visual objects among views on different displays. For instance, to visualize relationships among items on multiple displays, we can add a new display that holds relationship marks among items in multiple displays (Figure 6 (e)). Additionally, inspired by the group-level relationship marks, we can also consider approaches for showing relationships of data items on separate displays corresponding to one or more visual links from a source to multiple targets across multiple displays.

In summary, making design decisions by considering the in-between space and separate view context are particularly important if large displays and MDEs are used for visual analysis and sensemaking [73], [83]. In such analysis environments, users need appropriate strategies to understand and relate information distributed among separate displays and more seamlessly transition among different views on multiple displays during insight formation [84].

8 CONCLUSION

We discuss systematic design considerations for visualizing cross-view data relationships, with regards to both descriptive relationship aspects and usable visual context of multiple views. Based on these, the visualization design of cross-view data relationships can be understood as encoding desired relationship aspects by using available the visual

context of multiple views. We have analyzed a variety of design options, and made five design recommendations.

Despite proposing a systematic view of design considerations for cross-view data relationships, this work has four limitations. First, our notion of data relationship assumes relational data, but not all data in visualization is relational, or even discrete, so the designs that we have examined and discussed do not cover those cases. Second, we do not consider using multiple views in a 3D or immersive environment, which may lead to a broader set of design considerations with more advanced user interactions (e.g., ImAxes [85] and [86]). Third, our search of relevant designs is based on published papers using known exemplars and keyword search. Because of the small size of this group, even with recursive searches, our collected paper list is not exhaustive, potentially omitting relevant work. Last, although we have studied a variety of designs and analyzed their pros and cons, we have not performed much evaluation of these design considerations. Conducting user studies and expert interviews will be helpful to further evaluate these design considerations. However, we believe this work can inspire and guide future studies to further formalize a design space of visualizing cross-view data relationships.

ACKNOWLEDGMENTS

REFERENCES

[1] J. Zhao, "Interactive visual data exploration: A multi-focus approach," Ph.D. dissertation, Department of Computer Science, University of Tornoto, Jul 2015.

[2] A. Lex, M. Streit, E. Kruijff, and D. Schmalstieg, "Caleydo: Design and evaluation of a visual analysis framework for gene expression data in its biological context," in *IEEE Pacific Visualization Symposium*. IEEE, 2010, pp. 57–64.

[3] M. Sun and G. Convertino, "Interver: Drilling into categorical-numerical relationships," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2016, pp. 44–51.

[4] H. Zhang, M. Sun, D. Yao, and C. North, "Visualizing traffic causality for analyzing network anomalies," in *Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics*, 2015, pp. 37–42.

[5] J. Stasko, C. Görg, and Z. Liu, "Jigsaw: supporting investigative analysis through interactive visualization," *Information visualization*, vol. 7, no. 2, pp. 118–132, 2008.

[6] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan, "Exploratory analysis of time-series with chronolenses," *IEEE Trans. on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2422–2431, 2011.

[7] C. North and B. Shneiderman, "A taxonomy of multiple window coordination," Tech. Rep., 1997.

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

[9] J. Roberts, "State of the art: Coordinated & multiple views in exploratory visualization," in *International Conf. on Coordinated & Multiple Views in Exploratory Visualization*. IEEE, 2007, pp. 61–71.

[10] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky, "Guidelines for using multiple views in information visualization," in *Proceedings of the Working Conference on Advanced Visual Interfaces*. ACM, 2000, pp. 110–119.

[11] J. S. Yi, Y. ah Kang, and J. Stasko, "Toward a deeper understanding of the role of interaction in information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1224–1231, 2007.

[12] N. Boukhelifa, J. C. Roberts, and P. J. Rodgers, "A coordination model for exploratory multiview visualization," in *Proceedings of International Conference on Coordinated and Multiple Views in Exploratory Visualization*. IEEE, 2003, pp. 76–85.

[13] T. Pattison and M. Phillips, "View coordination architecture for information visualisation," in *Proceedings of the 2001 Asia-Pacific Symposium on Information Visualisation-Volume 9*. Australian Computer Society, Inc., 2001, pp. 165–169.

[14] R. Burtner, S. Bohn, and D. Payne, "Interactive visual comparison of multimedia data through type-specific views," in *Visualization and Data Analysis 2013*, 2013, p. 86540M.

[15] "In-spire," https://in-spire.pnnl.gov/.

[16] "Tableau software," https://www.tableau.com/.

[17] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[18] D. Park, S. M. Drucker, R. Fernandez, and N. Elmqvist, "Atom: A grammar for unit visualizations," *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[19] S. Huron, R. Vuillemot, and J.-D. Fekete, "Visual sedimentation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2446–2455, 2013.

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

[21] E. H.-h. Chi, "A taxonomy of visualization techniques using the data state reference model," in *IEEE Symposium on Information Visualization*. IEEE, 2000, pp. 69–75.

[22] J. C. Roberts, "Multiple view and multiform visualization," in *Visual Data Exploration and Analysis VII*, vol. 3960. International Society for Optics and Photonics, 2000, pp. 176–186.

[23] S. Knudsen and S. Carpendale, "View relations: An exploratory study on between-view meta-visualizations," in *Proceedings of 9th Nordic Conference on Human-Computer Interaction*, 2016, pp. 1–10.

[24] C. Viau and M. J. McGuffin, "Connectedcharts: explicit visualization of relationships between data graphics," in *Computer Graphics Forum*, vol. 31, no. 3pt4, 2012, pp. 1285–1294.

[25] C. Dunne, N. Henry Riche, B. Lee, R. Metoyer, and G. Robertson, "Graphtrail: Analyzing large multivariate, heterogeneous networks while supporting exploration history," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 1663–1672.

[26] M. C. Chuah and S. F. Roth, "On the semantics of interactive visualizations," in *Proceedings IEEE Symposium on Information Visualization'96*. IEEE, 1996, pp. 29–36.

[27] P. Koytek, C. Perin, J. Vermeulen, E. André, and S. Carpendale, "Mybrush: Brushing and linking with personal agency," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 605–615, 2018.

[28] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-lite: A grammar of interactive graphics," *IEEE Trans. on Visualization & Computer Graphics*, vol. 23, no. 1, pp. 341–350, 2017.

[29] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit, "Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2023–2032, 2014.

[30] J. Zhao, M. Glueck, F. Chevalier, Y. Wu, and A. Khan, "Egocentric analysis of dynamic networks with egolines," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 5003–5014.

[31] M. Sun, C. North, and N. Ramakrishnan, "A five-level design framework for bicluster visualizations," *IEEE Transactions on Visualization & Computer Graphics*, vol. 20, no. 12, pp. 1713–1722, 2014.

[32] H. Wu, M. Sun, P. Mi, N. Tatti, C. North, and N. Ramakrishnan, "Interactive discovery of coordinated relationship chains with maximum entropy models," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 12, no. 1, pp. 1–34, 2018.

[33] M. Sun, P. Mi, C. North, and N. Ramakrishnan, "Biset: Semantic edge bundling with biclusters for sensemaking," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 310–319, 2016.

[34] S. Cheng and K. Mueller, "The data context map: Fusing data and attributes into a unified display," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 121–130, 2016.

[35] C. North, "Toward measuring visualization insight," *IEEE computer graphics and applications*, vol. 26, no. 3, pp. 6–9, 2006.

[36] A. Endert, P. Fiaux, and C. North, "Semantic interaction for visual text analytics," in *Proc.of the SIGCHI Conf. on Human Factors in Computing Systems*. ACM, 2012, pp. 473–482.

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

[38] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers, "Visualizing sets and set-typed data: State-of-the-art and future challenges," in *Eurographics Conference on Visualization (EuroVis)–State of The Art Reports*, 2014, pp. 1–21.

[39] C. Andrews, A. Endert, and C. North, "Space to think: large high-resolution displays for sensemaking," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.   ACM, 2010, pp. 55–64.

[40] C. Weaver, "Building highly-coordinated visualizations in improvise," in *IEEE Symposium on Information Visualization*.   IEEE, 2004, pp. 159–166.

[41] C. North and B. Shneiderman, "Snap-together visualization: A user interface for coordinating visualizations via relational schemata," in *Proceedings of the Working Conference on Advanced Visual Interfaces*.   ACM, 2000, pp. 128–135.

[42] C. Weaver, "Cross-filtered views for multidimensional visual analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 2, pp. 192–204, 2010.

[43] C. Collins and S. Carpendale, "Vislink: Revealing relationships amongst visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1192–1199, 2007.

[44] M. Waldner, W. Puff, A. Lex, M. Streit, and D. Schmalstieg, "Visual links across applications," in *Proceedings of Graphics Interface*. Canadian Information Processing Society, 2010, pp. 129–136.

[45] B. Shneiderman and A. Aris, "Network visualization by semantic substrates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 733–740, 2006.

[46] P. Fiaux, M. Sun, L. Bradel, C. North, N. Ramakrishnan, and A. Endert, "Bixplorer: Visual analytics with biclusters," *Computer*, vol. 46, no. 8, pp. 90–94, 2013.

[47] M. Wertheimer, "Gestalt theory." 1938.

[48] C. Ware, H. Purchase, L. Colpoys, and M. McGill, "Cognitive measurements of graph aesthetics," *Information visualization*, vol. 1, no. 2, pp. 103–110, 2002.

[49] S. Kieffer, T. Dwyer, K. Marriott, and M. Wybrow, "Hola: Human-like orthogonal network layout," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 349–358, 2015.

[50] A. Lex, M. Streit, C. Partl, K. Kashofer, and D. Schmalstieg, "Comparative analysis of multidimensional, quantitative data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1027–1035, 2010.

[51] A. Lex, H.-J. Schulz, M. Streit, C. Partl, and D. Schmalstieg, "Visbricks: multiform visualization of large, inhomogeneous data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2291–2300, 2011.

[52] M. Steinberger, M. Waldner, M. Streit, A. Lex, and D. Schmalstieg, "Context-preserving visual links," *IEEE Trans. on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2249–2258, 2011.

[53] H. Chung and C. North, "Savil: cross-display visual links for sensemaking in display ecologies," *Personal and Ubiquitous Computing*, pp. 1–23, 2017.

[54] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE Trans. on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 91–100, 2016.

[55] M. Waldner and D. Schmalstieg, "Collaborative information linking: Bridging knowledge gaps between users by linking across applications," in *IEEE Pacific Visualization Symposium*.   IEEE, 2011, pp. 115–122.

[56] T. Geymayer, M. Steinberger, A. Lex, M. Streit, and D. Schmalstieg, "Show me the invisible: visualizing hidden content," in *Proceedings of the 32nd annual ACM Conference on Human Factors in Computing Systems*.   ACM, 2014, pp. 3705–3714.

[57] A. Lex, M. Streit, H.-J. Schulz, C. Partl, D. Schmalstieg, P. J. Park, and N. Gehlenborg, "Stratomex: visual analysis of large-scale heterogeneous genomics data for cancer subtype characterization," in *Computer graphics forum*, vol. 31, no. 3pt3, 2012, pp. 1175–1184.

[58] M. Streit, S. Gratzl, M. Gillhofer, A. Mayr, A. Mitterecker, and S. Hochreiter, "Furby: fuzzy force-directed bicluster visualization," *BMC bioinformatics*, vol. 15, no. S6, p. S4, 2014.

[59] B. Alper, N. Riche, G. Ramos, and M. Czerwinski, "Design study of linesets, a novel set visualization technique," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2259–2267, 2011.

[60] C. Collins, G. Penn, and S. Carpendale, "Bubble sets: Revealing set relations with isocontours over existing visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1009–1016, 2009.

[61] N. H. Riche and T. Dwyer, "Untangling euler diagrams," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1090–1099, 2010.

[62] K. Dinkla, M. J. van Kreveld, B. Speckmann, and M. A. Westenberg, "Kelp diagrams: Point set membership visualization," in *Comp. Graphics Forum*, vol. 31, no. 3pt1, 2012, pp. 875–884.

[63] W. Meulemans, N. H. Riche, B. Speckmann, B. Alper, and T. Dwyer, "Kelpfusion: A hybrid set visualization technique," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 11, pp. 1846–1858, 2013.

[64] J.-F. Im, M. J. McGuffin, and R. Leung, "Gplom: the generalized plot matrix for visualizing multidimensional multivariate data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2606–2614, 2013.

[65] I. Boyandin, E. Bertini, P. Bak, and D. Lalanne, "Flowstrates: An approach for visual exploration of temporal origin-destination data," in *Computer Graphics Forum*, vol. 30, no. 3, 2011, pp. 971–980.

[66] Y. Yang, T. Dwyer, S. Goodwin, and K. Marriott, "Many-to-many geographically-embedded flow visualisation: an evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 411–420, 2017.

[67] J. Zhao, M. Sun, F. Chen, and P. Chiu, "Bidots: Visual exploration of weighted biclusters," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 195–204, 2018.

[68] M. Sun, J. Zhao, H. Wu, K. Luther, C. North, and N. Ramakrishnan, "The effect of edge bundling and seriation on sensemaking of biclusters in bipartite graphs," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 10, pp. 2983–2998, 2018.

[69] N. Henry, J.-D. Fekete, and M. J. McGuffin, "Nodetrix: a hybrid visualization of social networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1302–1309, 2007.

[70] M. Sun, D. Koop, J. Zhao, C. North, and N. Ramakrishnan, "Interactive bicluster aggregation in bipartite graphs," in *2019 IEEE Visualization Conference (VIS)*.   IEEE, 2019, pp. 246–250.

[71] D. Hienert, B. Zapilko, P. Schaer, and B. Mathiak, "Vizgr: linking data in visualizations," in *International Conf. on Web Information Systems and Technologies*, 2011, pp. 177–191.

[72] N. Boukhelifa and P. J. Rodgers, "A model and software system for coordinated and multiple views in exploratory visualization," *Information Visualization*, vol. 2, no. 4, pp. 258–269, 2003.

[73] R. Langner, T. Horak, and R. Dachselt, "Vistiles: Coordinating and combining co-located mobile devices for visual data exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 626–636, 2018.

[74] S. Bjork and J. Redstrom, "Redefining the focus and context of focus+context visualization," in *IEEE Symposium on Information Visualization*.   IEEE, 2000, pp. 85–89.

[75] W. Javed and N. Elmqvist, "Exploring the design space of composite visualization," in *Visualization Symposium (PacificVis), 2012 IEEE Pacific*.   IEEE, 2012, pp. 1–8.

[76] E. Zgraggen, R. Zeleznik, and S. M. Drucker, "Panoramicdata: Data analysis through pen & touch," *IEEE Trans. on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2112–2121, 2014.

[77] M. A. Yalcin, N. Elmqvist, and B. B. Bederson, "Aggreset: Rich and scalable set exploration using visualizations of element aggregations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 688–697, 2016.

[78] M. A. Yalçın, N. Elmqvist, and B. B. Bederson, "Keshif: Rapid and expressive tabular data exploration for novices," *IEEE Transaction on Visualization and Computer Graphics*, 2017.

[79] J. Heer and M. Bostock, "Crowdsourcing graphical perception: using mechanical turk to assess visualization design," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 203–212.

[80] C. Andrews and C. North, "Analyst's workspace: An embodied sensemaking environment for large, high-resolution displays," in *IEEE Conference on Visual analytics Science and Technology*.   IEEE, 2012, pp. 123–131.

[81] R. Sadana, T. Major, A. Dove, and J. Stasko, "Onset: A visualization technique for large-scale binary set data," *IEEE Transaction on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 1993–2002, 2014.

[82] J. Fekete, D. Wang, N. Dang, A. Aris, and C. Plaisant, "Overlaying graph links on treemaps," in *Proceedings of the IEEE Symposium on Information Visualization Conference Compendium (InfoVis' 03)*. IEEE, 2003, pp. 82–83.

[83] H. Chung, C. North, S. Joshi, and J. Chen, "Four considerations for supporting visual analysis in display ecologies," in *IEEE Conference on Visual Analytics Science and Technology*. IEEE, 2015, pp. 33–40.

[84] H. Chung, C. North, J. Z. Self, S. Chu, and F. Quek, "Visporter: facilitating information sharing for collaborative sensemaking on multiple displays," *Personal and Ubiquitous Computing*, vol. 18, no. 5, pp. 1169–1186, 2014.

[85] M. Cordeil, A. Cunningham, T. Dwyer, B. H. Thomas, and K. Marriott, "Imaxes: Immersive axes as embodied affordances for interactive multivariate data visualisation," in *Proc. of ACM Symposium on User Interface Software and Technology*. ACM, 2017, pp. 71–83.

[86] A. Prouzeau, A. Lhuillier, B. Ens, D. Weiskopf, and T. Dwyer, "Visual link routing in immersive visualisations," in *Proceedings of the ACM International Conference on Interactive Surfaces and Spaces*, 2019, pp. 241–253.

Tianyi Li is an Assistant Professor with the Department of Computer Science, Loyola University Chicago. She designs and develops systems for computer-supported cooperative work. Her most recent research is about crowdsourced sensemaking, to scaffold collective intelligence of novice crowds for tasks such as intelligence analysis.
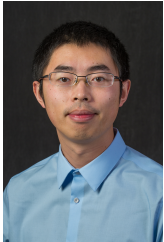
Maoyuan Sun is an assistant professor with the Department of Computer Science, Northern Illinois University. His research falls in the areas of visual analytics, information visualization, human-computer interaction and human-centered machine learning, with a variety of applied domains, including STEM education, intelligence analysis, business intelligence and cyber security.

Akhil Namburi is a graduate student with the Department of Computer Science, Northern Illinois University. His research focuses on creating usable, interactive user interface for supporting sensemaking of big data.

Haeyong Chung is Assistant Professor with the Department of Computer Science, the University of Alabama in Huntsville. His research focuses on the design and development of visual analysis techniques and tools that leverage mixed displays including mobile devices and tiled display walls.

David Koop is an assistant professor with the Department of Computer Science, Northern Illinois University. His research concerns provenance, data science, and visualization.
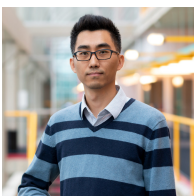
Jian Zhao is an assistant professor in the Cheriton School of Computer Science at the University of Waterloo, where he directs the WatVis (Waterloo Visualization) group. His research interests include information visualization, human-computer interaction, and data science. His work contributes to the development of advanced interactive visualizations that promote the interplay of human, machine, and data.