An Empirical Guide for Visualization Consistency in Multiple Coordinated Views

Shaocong Tan[1] * Chufan Lai[1] † Xiaolong (Luke) Zhang[2] ‡ Xiaoru Yuan[1,3] §

1) Key Laboratory of Machine Perception (Ministry of Education), and School of Intelligence Science and Technology, Peking University, China
2) College of Information Sciences and Technology, Pennsylvania State University, USA
3) National Engineering Laboratory for Big Data Analysis and Application, Peking University, China

ABSTRACT

Visual analytic systems usually provide multiple coordinated views (MCVs) to support data analysis and exploration. Coordination in visual graphics plays an important role in facilitating comprehensive analytical tasks, such as data comparison and cognitive inference. However, individual views in MCVs are probably designed for a specific purpose based on a particular type of data, and insufficient consideration of the intricate relationships among views may lead to inconsistency in visual representation and user interaction across different views. To better understand the inconsistency issues in MCVs and their impacts on user behaviors, this paper reports a study on the analysis and classification of visualization inconsistency based on the reviews of interactive visualization designs and visual analytic systems, and the interviews with stakeholders. We find that inconsistencies are prevalent in MCVs and frequently lead to misleading or even incorrect results. We classify the discovered inconsistencies based on a coordination model of MCVs, and develop an empirical guide for systematic and efficient visualization consistency checking in the design, implementation, and evaluation stage.

Index Terms: Human-centered computing—Visualization—Empirical studies in visualization;

1 INTRODUCTION

With the growing popularity of visualization in our daily lives and work, multiple coordinated views (MCVs) have become a common design element used to enhance expressiveness and analytics. MCVs are effective at revealing data characteristics across various dimensions, associations between different entities, and can even facilitate the discovery of previously unseen relationships and patterns [25]. The coordination of views is closely tied to consistency, which is crucial as users are often presented with a set of views simultaneously [31], each illustrating different operation states and corresponding data. Well-designed MCVs can seamlessly integrate independent parts into a larger purposeful whole, while poorly designed ones may introduce additional complexity and result in a steeper learning curve [42]. Consistency is therefore an essential measure of performance in MCVs such as *small multiples*, enabling users to establish visual connections between views, make correct inferences about the data, and facilitate tracking and comparisons.

However, creating consistent MCVs that establish proper connections can be a challenging task. While numerous guidelines and criteria exist for making visualizations effective, attractive, and memorable, most focus on single views and lack guidance on achieving consistency in MCVs. This can result in designers being unaware of inconsistency issues that can impact the creation of correct mental models. From our observations of many visualization designs, maintaining consistency can be particularly error-prone in MCV design. In contemporary visualization projects, multiple people often collaborate on design and implementation, making inconsistency issues more likely to arise when different people focus on their own separate modules. As the number of views and complexity of coordination increase, systematically detecting and correcting inconsistency issues becomes an even more daunting task [37].

Although designers may sometimes need to compromise consistency in favour of other design considerations such as functionality and aesthetics, it has been argued that consistency is one of the most important heuristics for ensuring effectiveness and can help to explain a wide range of usability problems [15]. Inconsistencies can have a negative impact on the effectiveness of visualizations. Inconsistent visual encoding in the user interface can cause cognitive confusion and impede learning [28]. Inconsistent system states presented by different views can get users disoriented in exploration and fail to obtain the desired information. Even worse, inconsistencies can lead to the faulty inference, causing users to misinterpret data and make potentially disastrous decisions. This problem certainly deserves attention, but to our knowledge, there are currently no satisfactory methods or tools available to address it.

This paper reports our research on developing an empirical guide for visualization consistency in MCVs. The research is based on the review of 898 interactive visualizations created by students in two courses from 2012 to 2022, the tracking of 14 visual analytic systems completed by experienced people, and the interview with 15 stakeholders. The contributions of our work are as follows:

- **C1:** Deriving consistency guidelines in MCVs based on existing literature and real-world practice.
- **C2:** Investigating common causes of inconsistency issues for different types of visualization stakeholders.
- **C3:** Identifying prevalent inconsistency issues in visualization and visual analytics systems that use MCVs.
- **C4:** Developing a coordination model of MCVs to categorize the causes of inconsistency issues systematically.
- **C5:** Providing a practical guide for efficient visualization consistency checking in MCVs.

Our research can help reduce inconsistency in MCVs by providing refined definitions of consistency and elaborations of inconsistency issues. We also invited 20 visualization stakeholders for evaluation to demonstrate that people from diverse backgrounds could benefit from our work. By providing step-by-step instructions, people can increase the efficiency of integrating consistency guidelines into visualization practices, avoiding mindless and disorganized checking.

2 RELATED WORK

For our focus on consistency in MCVs, this section reviews related research on coordination models, user interaction in MCVs, and consistency in visualization design.

*e-mail:tanshaocong0108@gmail.com
†e-mail:chufan.lai.1990@gmail.com
‡e-mail:lzhang@ist.psu.edu
§e-mail:xiaoru.yuan@pku.edu.cn (corresponding author)

2.1 Coordination Models

Coordination refers to the connections between views that can make interrelated views more informative than what any constituent views can express alone [5,25,31,33]. Coordination models concerning the design of MCVs help describe the dependencies between coordinated views and improve understanding of the state management of views. Existing models can be categorized into two types.

Data-oriented models primarily use the data involved in MCVs as the foundation. The Snap-together model [26] connected views based on the relational data model and associated them by joining data through database-like operations. The Model-View-Controller (MVC) architecture [27] facilitates the management of multi-view system by decoupling data logic from visual encoding. The dataflow model [5] used such abstract objects in data analysis as the base for view coordination. By focusing on the intrinsic relationships among data tables, these models offer an effective way to link views logically. Interaction-oriented models emphasize the role of user interaction in the coordination of views. Weaver et al.'s model [45] defined interactive dependency between views based on dynamic data properties during user exploration. *Nebula* [10] formalizes coordination as user- and coordination-triggered interaction accompanied by data transformation to manage the potential inconsistency of the backing data between views. These models are more flexible in supporting dynamic visual analytic behaviours.

With increasingly complex coordination, the structure that ensures consistent coordination among views becomes more and more important for successful data exploration. These coordination models lay the foundation for our research. Through analyzing multiple visualization designs, this paper extends existing coordination models and abstracts coordination from a more general perspective to describe inconsistency issues in visualization with MCVs.

2.2 User Interaction in MCVs

User interaction, an essential component in visualization-based data exploration, is often a driving factor for coordination. In MCVs, the views are updated dynamically based on user actions to provide users with different perspectives and gain insight [31].

Interaction-driven coordinated views can assist users in understanding complex data and conducting sophisticated analyses [13]. For example, users can locate anomalies, compare similarities or differences, and isolate and aggregate data subsets through interactions. When users perform tasks that concern or influence data involved in different views, the coordinated views should update synchronously in response to user queries. Common interaction scenarios include using *brushing & linking* to explore data relationships holistically [4,31], applying the *dynamic query* to identify relevant data sets in different views [35], or *zooming & panning* in one view to control the scope of data in all linked views [43].

Although user interaction is crucial in visual analytics, ignoring some design guidelines can lead to distraction and misunderstandings. Inconsistency is one of the seven costs of user interaction in visualization, as identified by Lam et al. [20]. The unpredictable effects of interaction on the system state may introduce unnecessary cognitive load, hinder user efforts to construct mental models, and ultimately reduce the efficiency of data exploration. User interactions that are challenging to comprehend and operate can be harmful in MCVs [12]. Therefore, it is a pressing issue in consistency studies.

2.3 Consistency in Visualization Design

Consistency is an important consideration in user interface design, and it has been recognized by the visualization community for many years. Consistency not only facilitates comparison and inference of visualization but also prevents confusion or misinterpretation among viewers [42]. In fact, consistency is listed as one of the top ten heuristics for evaluating the usability of visualization systems [15].

A diverse body of research has touched on the issue of consistency in visual user interfaces that involve multiple views [28,32]. The "Rule of consistency" [42] urged attention to the consistency of system state and user interface. Forsell et al. [15] interpreted consistency as "the way design choices are maintained in similar contexts and are different when applied to different contexts". Wu et al. presented *DIEL* [48] to coordinate asynchronous events over distributed data and maintain consistent user experiences in design. The works by Qu et al. [29] adopted a similar concept for encoding-specific rules and described how it integrated with other design considerations, e.g., *xy* and color encoding [30]. Recently, Kristiansen et al. introduced Semantic Snapping [19], which quantifies two types of inconsistency (hallucinator and confuser) by evaluating the algebraic relations of grouping, channel, data mapping, and visual output among views. These approaches are typically proposed for limited visual encodings in static visualizations.

Previous literature on visualization consistency has largely offered abstract and sweeping concepts, which may not effectively guide practitioners in avoiding inconsistency issues. While some works have explored consistency in specific scenarios or visual encodings, there is a need for a more comprehensive summary that can apply to interactive visualization with multiple coordinated views.

3 Terminology

To enhance clarity, we introduce the related terms, i.e., MCVs and system state, and define key concepts in our research, i.e., consistency and inconsistency issues in MCVs, based on existing literature and our preliminary study presented in Appendix A.

MCVs is a type of visualization design with two or more views. It dynamically updates some views in response to user interaction [10,45]. In multi-view visualization, coordination among views is implied as the default [31]. The view here is a broad concept. Even the titles, legends, and widgets can be regarded as a view and should be updated consistently during user interaction.

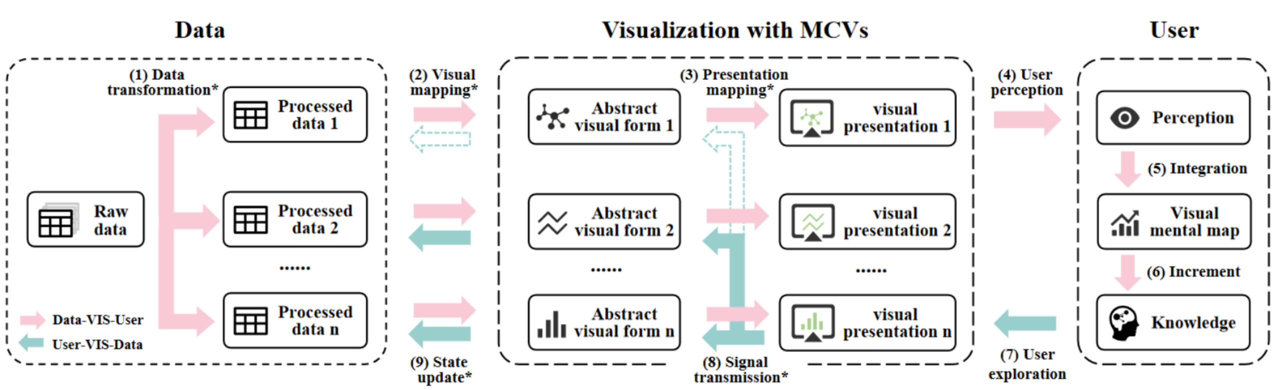


Figure 1: The coordination model of MCVs. It represents the cycle of visualization with the consideration of user interaction, following Van Wijk's simple model of visualization [39]. During an interaction, *User* makes specifications for one or more views (*VIS*), which triggers corresponding transformations in the *Data*. The processed *Data* is then mapped into time-varying *VIS* that can be perceived by the *User* in multiple views.

We developed a general coordination model for MCVs, depicted in Fig. 1, with reference to previous visualization models [17, 39] and coordination models [5, 10]. This model represents the complete cycle of visualization, including visualization construction and coordination. Note that in MCVs, multiple visualizations can be derived from the same data source, while different data sources can generate a visualization [12]. For generality, we are not concerned with the practical data relationships and their integration approaches. Therefore, the term "processed data" represents the resulting data for visualization, which can come from one or more data sources.

System state consists of data state and user viewpoint [42], and is used to describe view coordination and update behaviour of MCVs. User interactions are translated into system states that drive visualization updates. An interaction is "the interplay between a person and a data interface involving a data-related intent" [14]. Data state characterizes the expected manners of each data unit in visualization at initialization or after an interaction, indicating which data unit should be shown or hidden, highlighted or faded. User viewpoint includes the orientation of views and the level of detail (LOD), like specifying the camera position, the viewing direction, the frustum size, and the LOD indicator when shooting [44]. The system state is a key factor affecting consistency, as noted by Wang et al. [42].

Consistency in MCVs is one of the important design considerations for multi-view visualization. In Sect. 2.3, we introduced the definition of consistency from existing works [15, 19, 29, 30, 42, 48]. We follow their common ideas and extend the guidelines to a wider context of interactive MCVs based on the preliminary study.

G1: The same data field should be encoded in the same way. Different data fields should be encoded differently.

G2: The same data relationship should present in the same way. Different data relationships should present differently.

G3: The same functionality should have the same affordance. Different functionalities have different affordances.

G4: The coordinated views should update synchronously and display the same system state in response to user interaction.

G5: If the above is not desired, use instructions or perceptual cues to make relationships among views evident to users.

G1 concerns the visual marks and channels of data fields, which stipulates to avoid distracting users by unnecessary visual changes when the data field across views remains constant and the different data fields between individual views should be visually distinguishable [29]. For example, the same fields should be rendered at the same scale, and different fields are not expected to use the same or too similar color palettes. This guideline helps ensure that any changes in the data field between visualizations are easily noticeable and perceivable, reducing the perceptual workload of tracking the equivalent elements and distinguishing different data fields [30].

G2 focuses on the data relationships such as correlation, co-occurrence, and hierarchy, which are important aspects of visual analytics. However, presenting the same relationship in different views with varying visual connections or mapping schemes not only occupies extra visual channels but also imposes a cognitive burden on users to recognize the same relationship. On the other hand, indistinguishable encoding schemes can cause users to conflate different data relationships and draw false inferences. Consistent presentation of data relationship is helpful in reducing the risk of misinterpretation and minimize confusion in relational reasoning.

G3 is data independent and designed for visual presentations of interaction capabilities, such as selection or navigation functionality like *zooming & panning*. In multi-view interfaces, views with equivalent functionality should have the same affordances to make operations predictable and enhance system usability [42]. The consistent affordance of widgets with similar features can smooth the learning curve and prevent users from getting disconcerted by unexpected context switches [12]. On the other hand, distinguishing different functionalities through unique appearances can help users avoid useless attempts and promote exploration efficiency.

G4 is a guideline for user interaction and view coordination in MCVs. When users interact with the system, consistent data transformation and visual updates such as synchronized highlighting and navigating can help them establish visual connections and facilitate comparisons across views. In contrast, users are likely to obtain wrong data insights and draw incorrect conclusions due to inconsistent system states and inconsistent data presentations [42].

G5 is an additional suggestion that applies when consistency guidelines cannot be guaranteed in MCVs. Although consistency is often important, there are situations where adhering to the above guidelines is not desirable or achievable. Practical requirements and other design considerations may necessitate sacrificing certain consistency. In such cases, we suggest designing views that enable users to discern the decoupling relationships between them, even if they are non-compliance with the consistency guidelines.

These guidelines are applicable across various data types and visualization tasks, making them compatible with a wide range of user activities, including consumption, search, and querying [6].

Inconsistency issues in MCVs arise when any of the consistency guidelines are violated, leading to unnecessary discrepancies in different views. Inconsistency is a major obstacle to the effectiveness of MCVs. Based on the analysis of numerous real-world cases and the interviews with stakeholders, we have identified many inconsistency issues in visualization practice and categorized them into the five crucial links of the coordination model, as shown in Table 1. The violations of **G1** and **G2**, which pertain to the visual presentations of data fields and data relationships, can occur when inconsistent operations are performed in the processes from raw data to visualization (links 1-3) or when there are inconsistent changes to sysyem states (link 9). As **G3** is targeted at data-independent presentations, its violations can occur when specifying visual details in the presentation mapping (link 3). For violation of **G4**, there can exist inconsistency issues between views in each stage of coordination (links 8-9) and visualization updates (links 1-3) that cause the coordinated views to show inconsistent presentations of identical system states. The violations of **G5** occur when consistency guidelines are not followed at some stages, and instructions or perceptual cues are not provided on the user interface. In Sect. 4, we will discuss the inconsistency issues by category in more depth with concrete examples.

4 Classification of Inconsistency Issues in MCVs

This section explains each category of inconsistency issues in detail, starting with visualization construction and ending with interaction feedback, through real-world cases collected in our research.

Phase	Stage	Link	Operations	Potential Violations
Visualization Construction & Updates	Data-to-Data	(1) Data transformation	Derive processed data for visualization from raw data.	G1, G2, G4, G5
	Data-to-VIS	(2) Visual mapping	Map processed data into visual structures of views.	G1, G2, G4, G5
	VIS-to-User	(3) Presentation mapping	Generate fully-specified MCVs and render for users.	G1, G2, G3, G4, G5
Coordination	User-to-VIS	(8) Signal transmission	Deliver user specification to each coordinated view.	G4, G5
	VIS-to-Data	(9) State update	Translate action sequences and update system states.	G1, G2, G4, G5

Table 1: Five stages in the coordination model that are prone to inconsistency issues, along with potential violations of consistency guidelines.

4.1 Inconsistency in Data-to-Data

Inconsistency in Data-to-Data refers to the inconsistent data transformation in Fig. 1 (1). In this stage, the raw *Data* is transformed into processed *Data* suitable for specific visualization in different views [17]. However, the inconsistent data processing operations between views can derive inconsistent entities of the same data fields or produce inconsistent statistical values of the same entity in MCVs, leading to inconsistent presentations violating **G1**, **G2**, **G4**, or **G5**. Such inconsistencies can hinder users' ability to associate corresponding data entities or obtain proper data relationships across views, potentially resulting in erroneous conclusions. We will delve into common inconsistency issues in data transformation, focusing on basic attribute types and common operations [13, 23].

Namespace conversion for categorical attributes is often performed in inconsistent manners. Typically, the original variable is transformed into a format that is appropriate for display, taking into account the chart layout, available encoding resources, and other requirements. But two common issues that may violate **G1** or **G2**.

Aliases. The same variable may be processed into different abbreviations or retains its full names, as shown in Fig. 2 with the inconsistent expression of *months* in two axes. It is challenging for users to build visual connections and make comparisons across view.

Clipping. The variable names may be cut off in different ways, such as direct truncation, appending ellipses or other suffixes at varying positions, which can impede users from tracking the same variable and observing its features in multiple dimensions.

Mapping function for numerical attributes, which accepts a value in a specific *domain* and return a corresponding value in another *range*, is often implemented inconsistently, violating **G1** or **G2**. Users may read the inconsistent expressions of the equivalent value from MCVs, which can lead to confusion and misunderstanding.

Scale. Different views may adopt inconsistent scales, such as linear, logarithmic, or exponential scales, without sufficient instruction or hint. In Fig. 10, the upper and lower views use different scales, which can lead to incorrect results that deviate significantly from the actual values if unit indications are lost, violating both **G1** and **G2**.

Weights and measures. The measures may differ in references or multiples of the units, e.g., *m* and *km*. In Fig. 3, the numbers surrounded by ellipses are represented with inconsistent weights and measures, requiring extra effort from users to translate them.

Precision. Inconsistent numeric types or precisions can also lead to confusion. In Fig. 5, two bar charts display integer and floating numbers inconsistently, making it unintuitive that "KTA" and "CA" are actually correspond and the count is equal to each other.

Special handling for particular cases is carried out in different approaches, causing the violation of **G1** or **G2**. Below, we outline three common special cases that arise in data transformation.

For missing values, it is common practice to convert them to other values. However, the mapping to replacement values can sometimes be inconsistent across views. In the brown box of Fig. 4, the histogram treats the missing value as the individual symbol "Unknown", while the cartogram and ranking bar chart assign default values "<=1K" and "Others", respectively. In particular, the histogram of "District Count" skips counting (or filters out) the data entries with unknown authors. These varying treatments can lead to inconsistent distributions and erroneous statistical conclusions.

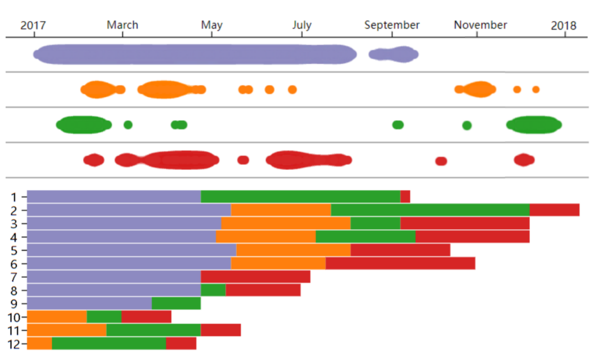


Figure 2: Inconsistent transformed namespaces and inconsistent axis directions for *month* make it difficult to observe two charts jointly.

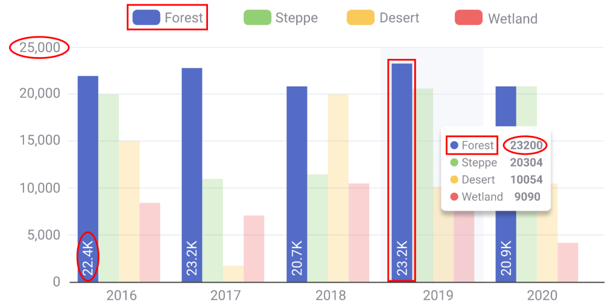


Figure 3: Inconsistent mapping functions and inconsistent visual marks. The needless discrepancies between the notations of numbers and inconsistent marks of the same variable may reduce readability.

For zero or infinite values, views may adopt varying approaches for handling or replacing these values. In Fig. 4, red boxes, the ranking bar chart specifies a minimum value to demonstrate the presence of data variables, while the histograms choose to keep them practical. These inconsistent treatments can lead to inconsistent data presentation in MCVs and increase the cognitive burden on users.

For extreme values, different methods may be used to handle them in MCVs. For example, the outliers may be intercepted by a limited range, compressed or expanded near the boundary, or mapped to specific intervals by non-linear functions. However, the mixed use of these methods can lead to inconsistent data visibility and hinder the observation of different attributes across views in MCVs.

Binning & Grouping are general data transformations used to categorize data and derive aggregation statistics. Their inconsistency issues can violate **G1** or **G2** and result in the different statistics objects across views, affecting users to build visual connections between corresponding objects and observe them in coordination.

For numerical attributes, inconsistent binning functions can cause the same values to be partitioned into different intervals, due to variations in bucket sizes, such as bin[0, 5], bin[0, 5), bin[0, 10]. In Fig. 4, yellow boxes, the inconsistent bins between views seriously reduce the interface's usability and increase the risk of false inferences.

For categorical attributes, the same values may be grouped into different classes in MCVs. It can arise from variations in classification schemes and standards, such as city categories and pollution levels. The case in Fig. 4, brown box, also shows the inconsistent classification of "Others" and "Unknown" between views.

Derivation method specifies how to derive statistics from data, but inconsistent definitions or calculations for the same indicator can lead to inconsistent statistical results, violating **G1** or **G2**.

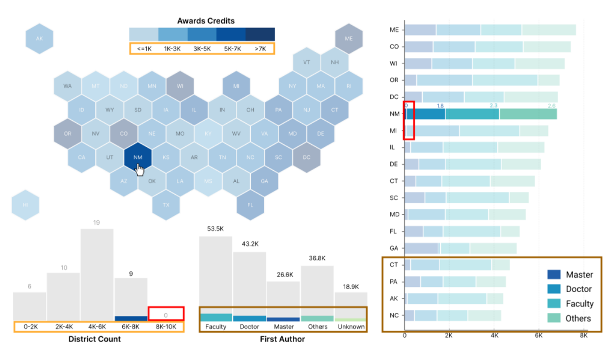


Figure 4: Inconsistent data binning & grouping, as well as inconsistent visual mappings. There are inconsistent groups of author types and inconsistent bins for counts among views. Also, the color schemes of the "Awards Credits" and "First Author" are too similar, violating **G1**.

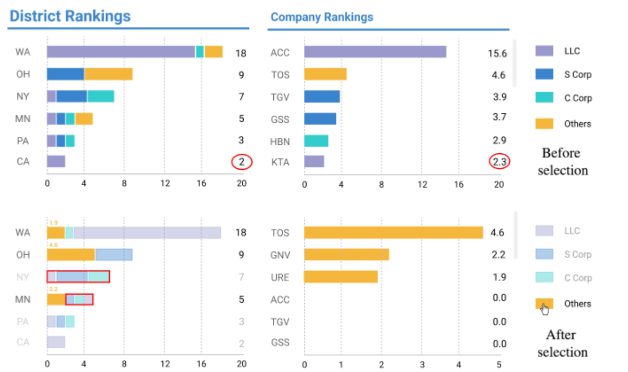


Figure 5: Inconsistent axis scales. After selecting a category, the "Company Rankings" takes the current maximum as a reference, resulting in inconsistent scales between views against **G1** and **G4**.

Definitions. Statistical indicators can be interpreted inconsistently across different views. Both domain-specific concepts and familiar operators may have varying definitions. According to the collaborators of Fig. 4, they used to mix up the definition of "Awards credits" as the total amount, the number of formal awards, and the weighted average in different views, leading to conflicting results in MCVs.

Calculations. The processes for computing identical indicators can be inconsistent. For instance, when calculating the similarity between data, two views may produce opposite results due to inconsistent setting of dimensional weights during projection. It is challenging to manage formulas and parameters in multiple views in case of data heterogeneity and collaborative implementation.

Filtering. The data subsets filtered for presentation may vary from view to view, reflecting inconsistent matching logic or inconsistent interpretation of data state in MCVs. This issue can result in inconsistent visualization that violates **G1**, **G2**, or **G4**.

Matching. Since filtering is a compound operation that is executed based on previous processing, inconsistency issues can be caused by other data transformations introduced earlier. For instance, some views could correctly answer a query for "United States", while others may produce fault response for preserving the transformed alias "USA". Statistics derived from inconsistent filtering subsets can cause users to form wrong insights through coordination.

Interpretation. Due to inconsistent understanding of the data state , views may filter out inconsistent data subsets in response to the same interaction, which violates **G4**. This presents the same problem as views not updating system states consistently, as in the cases shown in Fig. 11 and Fig. 6, which will be covered in Sect. 4.5.

Sorting involves organizing attributes or variables based on certain criteria, such as statistical values and lexicographic order. In cases where no other design considerations conflict with **G1**, **G2**, or **G4**, unnecessary inconsistent sorting can impose a cognitive burden on users who need to track and compare target objects across views.

Attribute Order. A consistent default ordering of data attributes across views naturally aligns with user cognition. Conversely, in the brown box of Fig. 4, the attribute of "First Author" is ranked inconsistently with that of the bar chart and the legend, which can increase users' visual work and cognitive load when comparing each attribute in two views, from sequential scan to disordered lookup. Similarly, in Fig. 5, the attribute order in rows with "Others" changes from to , which is inconsistent with legends and other rows. Users may need to reacquaint themselves with the order and experience difficulty when comparing attributes across rows.

Variable order. It is not advisable to sort variables inconsistently when views share dimensions or statistical indicators, unless there are specific reasons to do so. Apart from increasing cognitive load, such inconsistency can reduce the efficiency of user exploration. In Fig. 5, the left view does not follow the same ranking rule as the right view, which is to sort in descending order by "Others." It fails to provide a consistent answer to query about "Others" and obscures the relationship between regions and the companies within them.

Such inconsistency issues are common in multi-person collaborations and when separate data modules are used for each view. In these cases, updated logic may be changed in partial views without awareness of the other views. While inconsistent data transformations may be chosen intentionally to meet practical task requirements, it would be beneficial to have perceptual cues or instructions to notify users of these differences, as recommended by **G5**. Otherwise, users are probably confused and even misinterpret the data.

4.2 Inconsistency in Data-to-VIS

Inconsistency in Data-to-VIS refers to inconsistent visual mapping in Fig. 1 (2). Visual mapping aims to transform processed *Data* into the visual structure of *VIS*, which represents the initial and abstract form of the visualization. However, inconsistent mappings can create inconsistent graphical primitives and attributes, such as spatial substrates, marks, and graphical properties [7]. These issues can lead to inconsistent presentations of the underlying data in visualization construction and inconsistent visual changes in coordination through user interaction, which are against guidelines **G1**, **G2**, **G4**, or **G5**, making interpretation slow and error-prone [30]. In recounting such issues, we assume that the processed data in each view are consistent.

Visual mapping in visualization construction phase. We inherit two high-level constraints from **G1**: (1) Map the same data field to visual structure in the same way, and (2) Map different data fields to visual structures in different ways [15,29,30]. The constraints of the visual mapping of data relationships suggested by **G2** is comparable.

The same data field is mapped in different ways. Inconsistent visual structures can hinder users from tracking and observing the entities of interest. Users may be unable to create a visual connection across views, making multi-view coordination lose its value. Since a huge design space is available, we take four common manifestations here. In general, the inconsistent *colors*, *sizes*, or *shapes* mappings for the same field can make the interpretation inefficient and error-prone [29]. For example, if two shape channels encode the same nominal field, they should keep the same mapping from the nominal value to the symbol shape. In addition, the inconsistent *axis domains* can also increase the fallibility of user perception [30]. In Fig. 2, inconsistent axis direction for the same field let down the aligned distribution comparison in two views. In Fig. 5, inconsistent x scales between views can lead to a misleading visual presentation after selection. Some works have provided a more detailed classification, and evaluation of these issues [29,30]. Sometimes these guidelines must be weighed against other design considerations (e.g., chart layout, hue semantics, and whitespace) in specific scenarios, as the existing works examined [30,42]. We call for consistent visual mapping if the designers expect users to associate identical entities better and make comparisons across different views easier.

The different data fields are mapped in the same way. It is another appearance of inconsistent visual mapping where different data are mapped so alike. The inconsistency issues are prevalent in *colors*, *sizes*, or *shapes* mappings. The indistinguishable color encodings, e.g., the color for both statistics look exactly the same, leading to confusion or misinterpretation among viewers. If two shape channels don't encode different nominal variables with distinct glyphs, users may not perceive the difference between them and form correct inferences. According to previous works [29,30], different data should be encoded with different visual marks and channels when expecting users to ascertain the different entities and properties across charts to generate a consistent understanding of them.

Visual mapping in coordination and visualization update phase. During an interaction, the visual mapping should still adhere to the guidelines **G1** and **G2**, as in the construction phase. Meanwhile, the **G4** should also be considered. Since the visualization design space and user interactions are not exhaustive, we take the typical interaction scenario — *brushing & linking* as an example. It refer to the connection between views of the same data, such that exploration in one view affects the representation in the other, e.g., a selection causes highlighting in each connected data representation. However, the inconsistent highlighting results can imply false relationships among views and seriously affect user performance [42].

Object. The linking objects are inconsistent in MCVs, e.g., the highlighted or low-lighted visual elements. Generally, the visual marks in different views representing the same type of data should be changed in sync. But in Fig. 5, the axis names and value labels of rank lists have inconsistent low-lighted manners, raising users' visual workload to perceive the highlighting entities of the selected category in both views. Besides, the left view retains unselected bars by fading them, while the other removes the bars, causing inconsistent visibility and accessibility of objects between views.

Channel. The visual channels representing the same data state are inconsistent across views. In Fig. 11, the node-link diagram uses a unique border for the focus node, but the linking visual marks in other views do not change consistently. Also, highlighting with the bold and enlarged font is not applied consistently to each view. Similarly, the same problem arises with low-lighting manners. In Fig. 4, the histograms fill non-selected parts with a neutral color, while others use the transparency to distinguish highlighted elements, which can increase the cognitive load on users and make it difficult for them to track identical elements from view to view.

Scheme. The visual schemes supported for the same type of data state vary in different views. Some views may offer various schemes to highlight several features or distinct multiple linking degrees, but others lack partial schemes. For example, in Fig. 6, the scatter plot provides different schemes for filtered low-lighted, semi-highlighted, and highlighted points, but the cartogram merges the presentation of some states, leading to inconsistent correspondences between linking elements, not conform to the guidelines for visual mapping.

The analysis efficiency of the coordinated system depends heavily on the appropriate mapping from data to visual form. The consistent visual structure can make comparisons and inferences of data more accessible [42], enhancing the cognitive effort by maintaining design choices in similar contexts and diverging when applied to different contexts [15]. In contrast, inconsistency may confuse users' vision with superfluous matters in not following **G5**'s recommendations.

4.3 Inconsistency in VIS-to-User

Inconsistency in VIS-to-User refers to the inconsistent presentation mapping in Fig. 1 (3). Presentation mapping completes the visual structure into a fully-specified form by determining the details of visual variables such as style or layout. As the output of visualization, it establishes a bridge from *VIS* to *User*. However, inconsistencies in the visualization details may occur between views in the data representation, violating **G1**, **G2**, **G4**, or **G5**, as well as in the interface affordances, violating **G3**, **G4** or **G5**. Since inconsistencies may also arise in the previous stages, we stipulate that the visual structures are already constrained to be consistent.

Style is a collection of detailed properties for data-related visual elements, such as color indices, font sizes, and stroke widths. Its inconsistency issue refers to the inconsistent specification of encoding details for the same visual variable against **G1**, **G2**, or **G4**. For example, the low-lighted elements not assigned with the same transparency indices in Fig. 4 may muddy the sequential color scheme. The equivalent marks in the red box of Fig. 3 not assigned with equivalent rounding ratio may cause user discomfort and confusion due to the slight differences between views. We emphasize minimizing unnecessary inconsistency in visual style and maintaining consistent user perception of the same visual variable.

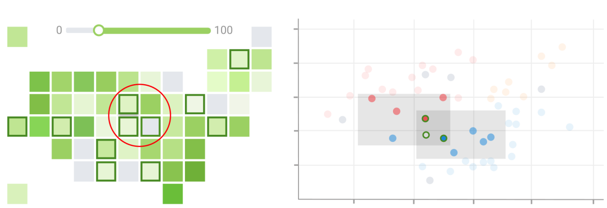


Figure 6: Inconsistent compatibilities and inconsistent highlighting schemes. After multiple brushing, the scatter plot and cartogram took the intersection and concatenation of the selected points, respectively.

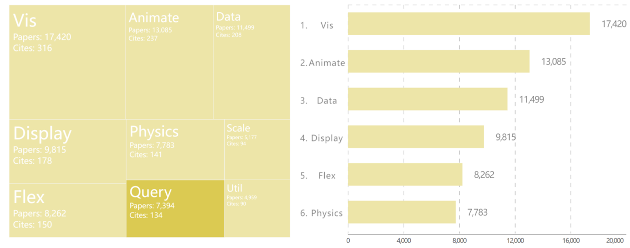


Figure 7: Inconsistent viewpoints. In the ranked list where the top-k elements are initially intercepted, the corresponding elements selected by treemap are not accessible due to the display thresholds. Navigation such as viewpoint shifting or element bubbling is expected.

Layout includes how visual elements are arranged and the spacing between them. The layout can imply the status of a visual element in the MCVs, especially its size and position [11]. The alignment and indentation can infer a hierarchy of elements, and area proportion denotes the importance of view [34]. In accordance with **G1** and **G2**, the visual elements at the same level, e.g., enumerated items, should be aligned to signify their equality, and the charts of parity are desirable with approximate size if the display resources are available, at least not quite imbalanced in the interface.

Navigation specifies the orientation and level of detail to view target data [44]. Inconsistent navigation may cause undesirable viewpoints and scope across views that violate guidelines **G1** and **G2**. The example in Fig. 7 shows a failed query of ranking through coordination which can be disconcerting to users. As for **G4**, linking views should show the same region as the user-triggered view did in response to interaction, e.g., *zooming & panning* [42].

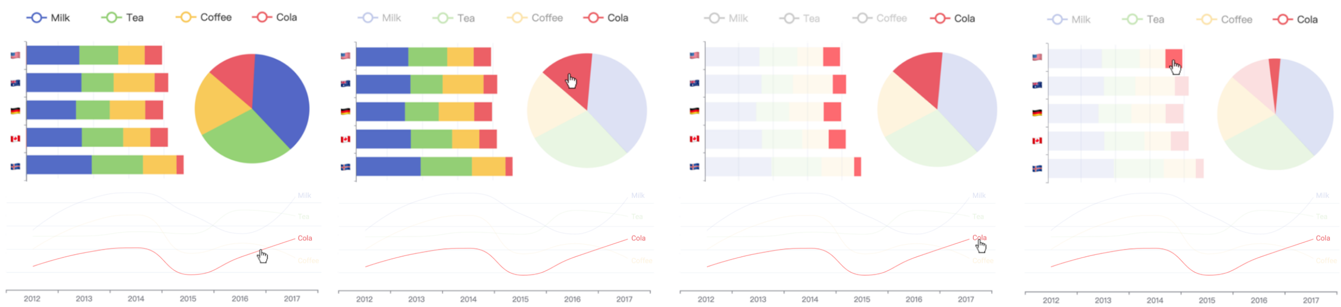
Transition is widely used to improve understanding and increase engagement in user interactions and view changes, e.g., rearrange and reload. However, inconsistent issues in violation of **G1**, **G2**, or **G4** often occur when applying transitions across views.

Animation with consistent semantic-syntactic mappings across data graphics for similar semantic operators can aid the understanding of data [16]. In contrast, inconsistent animation configurations may make users fail to establish visual connections and track items between views. For example, varying delay times may result in inconsistent beginnings to update the presented state, and varying duration times may cause the transitions out of sync.

Asynchronism. It is common to see inconsistent update progresses for coordination in MCVs due to asynchronous data access and various rendering time of visual presentation, making the displayed state against guideline **G4** in different views. In addition to the technique for coordinating asynchronous executions [48], adopting transitions for asynchronism such as progress bar ⌛ and loading indicator ↻, is helpful to maintain a consistent interface.

Decoration is non-coding graphical primitives used to promote the interpretation of visualization. But the inconsistent expressions violating **G3** and **G4** can cause negative or even harmful effects.

Instructions provided in data interface are often seen as inconsistent with the actual visual encodings or action modes of the presented data, such as the unit of value, legend, tooltip, and illustration. For example, in Fig. 10, the unit legend in the line chart did not update accordingly during user selection. Its mismatches with the actual scale make users prone to misinterpreting the values. Furthermore,



(a) Self-Loop (b) Choke (c) Ambiguity vs (a) (d) Divergence

Figure 8: Four categories of Inconsistency in User-to-VIS, including (a) Self-Loop, (b) Choke, (c) Ambiguity, (d) Divergence.

inappropriate instruction may lead users to false inferences. In Fig. 9, the illustration of the node with rimmed rings may mislead users that values are encoded by perimeter but not area.

Descriptions used to support details and insights elaboration such as titles, captions, and annotations may appear inconsistently. The font size, weight, and family can imply descriptions' juxtaposition, hierarchy, and importance. The inconsistent fonts of headers in Fig. 5 are inappropriate for equivalent juxtaposed views. It is unnecessary to use various fonts unless the text is emphasized or distinguished [7]. Ideally, the descriptions with equal terms have a consistent look to establish visual connections across different views.

Auxiliaries provided to facilitate visualization readability may introduce additional complexity with inconsistent appearances. For example, the lines served the same purpose, e.g., grid lines, but with varying forms may distract users from observing the material fact.

Hint. Visualization can guide user exploration by visual hints and information scents [47]. However, inconsistent hints in the user interface against **G3** and **G4** would instead introduce additional cognitive burden, even worse, mislead or discourage user exploration [42].

Hint for element should be provided consistently to avoid misculing users [42]. When visual elements with similar interactivity offer inconsistent hints, users may become confused and feel helpless in attempting further interaction. For example, data filters, navigation controllers, or configuration modifiers that support the same functionality, should offer consistent appearance and performance during tentative interaction. In Fig. 9, the subtitle linked to the homepage of *VIS Lab* has a different font from the title, which may not capture users' attention due to inconsistent perception. Sometimes, visual elements that do not provide consistent changes in response to trial interactions may indicate different interactivity. In the legend of Fig. 9, icons and labels that support the same interactive feature evoke inconsistent cursor reactions when mouse-over. In such cases, users may struggle to determine whether the change is intended or even noticed [20], and this can erode trust in the interface [20]. For

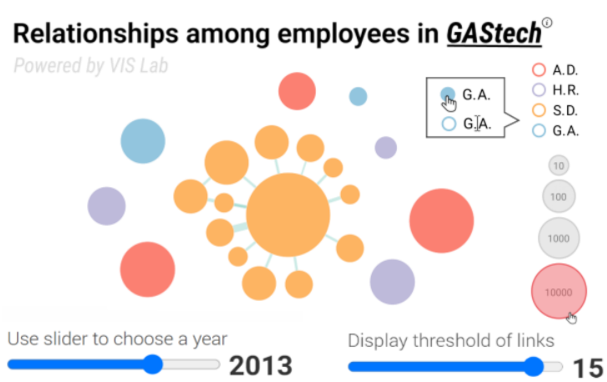


Figure 9: Inconsistent hints. Although both sliders have the same affordance, operating them leads to different results.

non-interactive elements that do not cause state updates or provide further information after an operation, it is not advisable to provide similar hints as interactive elements. Otherwise, users may engage in futile attempts to interact with these elements.

Hint for operation. In an interactive visualization, affordance is expected to suggest how to use the interactive elements based on their functionality and system state. Operations that trigger the same way should have a consistent visual clue or action guidance. For example, users may confuse about what operations the widgets target if they appear inconsistently as and in the interface. Such inconsistency introduces redundant visual variables and interferes with user cognition. For common interaction of *zooming & panning*, a consistent indication in MCVs such as , , or , can help prompt users about the upcoming effects of *zoom in* or *zoom out*. Otherwise, users are likely to get an inverse result against their intention, which is more frustrating than vague hints. On the other hand, users perceive the affordances in common as supporting comparable operations. Users may be guided to perform a false operation if their functionalities are disparate in reality [42]. The two sliders in Fig. 9 appear similar, but the "Display threshold" scale actually selects the part greater than "15" instead of the interval highlighted in blue. This can cause confusion among users and lead to fallible interpretations of the visual information.

Consistent visual presentation makes visualization with MCVs easier to learn and use, while inconsistency may lead to confusion or misinterpretation among users. It is usually overlooked by both novices and experienced people when combining individual views since the subtle differences between visual specifications take time to perceive. In the cases that consistent presentation cannot be achieved, it is recommended to use perceptual cues to make relationships and decoupling among multiple views apparent following **G5** [42].

4.4 Inconsistency in User-to-VIS

Inconsistency in User-to-VIS refers to inconsistent signal transmission in Fig. 1 (8). The signal serves as a messenger to deliver *User* exploration specification to different views (*VIS*), thereby linking user-triggered views with coordinated views [10]. However, inconsistent signal transmission can prevent coordinated views from receiving consistent commands synchronously, as shown by the dotted links in Fig. 1, leading to inconsistent state updates and interaction responses across views that violate **G4** or **G5**. Signal transmission involves signal *sending* and *receiving*, each of which may not execute or not execute consistently, resulting in four categories of issues.

Self-Loop. The signals are not broadcast to other views through coordination, i.e., not sent to each view. In Fig. 8 (a), the line chart as triggered view acquaint the interaction itself and own a distinctive data state than others. It is often the case for novices due to the lack of awareness of the coordination model and the technical burden of integrating visualization toolkits and libraries in MCVs.

Choke. The signals are not propagated to all coordinated views accordingly, i.e., not received by each view. In Fig. 8 (b), partial views with appropriate coordination consistently respond to the interaction with the updated data state while the bar chart still retains the previous state. It is often the case as Fig. 3 shown that designers focus on the coordination from legend to bars but ignore the consistent highlighting state in legend and tooltip when selecting a bar. According to our study, this issue is mainly for the non-consciousness that these views should be linked together in design, and the insufficient skill to achieve full coordination in implementation.

Ambiguity. The signals from the equivalent visual elements or views do not deliver the same information, i.e, not sent consistently. In Fig. 8 the line chart of (a) and (c), the line and its name label are supposed to have equal interactivity. However, when the mouse hovers over them, they both make the highlight reaction, but the signals transmitted to other views are different. It makes users difficult to understand the semantics of elements and their relationships, especially in response to the identical user operation. We often capture this situation when people collaborate to implement individual views without reaching a uniform communication standard.

Divergence. The received signals are not handled appropriately by some views, i.e., not received consistently. Similar to *Choke*, it will cause diverging system states in response to user interaction. In Fig. 8 (d), the bar chart and pie chart provide a filtered subset of the selected *Countries* and *Beverages* while the line chart shows the statistics for the full set. It is probably caused by not handling signals about *Country*. The issue frequently occurs in collaborative projects, as no uniform module handles the signals.

Inconsistent signal transmission can cause MCVs out of sync, which has to be resolved or avoided [5, 33]. Otherwise, users are likely to reason incorrectly with inconsistent states. However, the coordination is more intricate as the number of views increases. It is the most frequent inconsistency for novices and skilled people, as shown in Fig. 12. If the consistency between views cannot be implemented, the suggestion of **G5** should be followed.

4.5 Inconsistency in VIS-to-Data

Inconsistency in VIS-to-Data refers to inconsistent state updates in Fig. 1 (9). The system state represents the effect of signals in coordinated views (*VIS*) on *Data* [12]. Because the view does not spontaneously manipulate data but receives user specifications with signals, this stage can be considered the "User-to-Data" in case of consistent signal transmission. Inconsistent state updates indicate that the action sequences have been translated inconsistently to the system state in different views, which can cause inconsistent visual presentations in MCVs that violate **G1**, **G2**, **G4**, or **G5**. To ensure a consistent user mental map during the interaction, it is necessary to disambiguate the semantics of allowable actions and maintain consistent integration logic for action sequences. Since we cannot exhaust all user actions in this paper, we focus on the fundamental one — selection, which can be subjected to many actions such as highlighting, deleting, and navigating [44]. We identified three main factors when translating user selection to the system state.

Semantics establish the mapping from action to the system state, whose inconsistent results can affect the data transformation and visual mapping, leading to violations of **G1**, **G2**, and **G4**. If the action mappings are inconsistent across views, e.g., focus and association, poor interpretation and learnability of MCVs may result.

Focus performed by user selection may exist inconsistent semantics across views. For example, there are two common pinning strategies, as some imply the focus on the selected category, while others imply the elimination. But their mixture use in an interface can easily cause false interaction. There is another practical case in Fig. 10. Since the views offer inconsistent responses to focus, the chaos focus may confuse users during exploration.

Associations are not consistently derived from the selected entities in different views, e.g., co-occurrence and cooperation relationship. In Fig. 11, data states of associated keywords were not updated

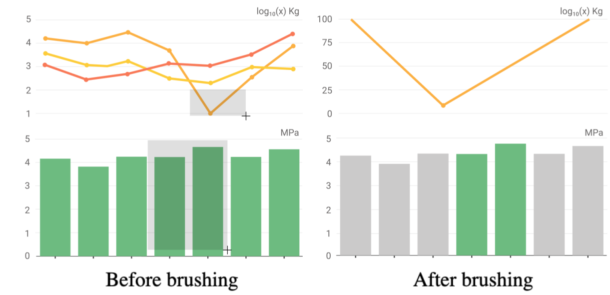


Figure 10: Inconsistent semantics of brushing. The selections trigger navigation and highlighting respectively, without any perception cue.

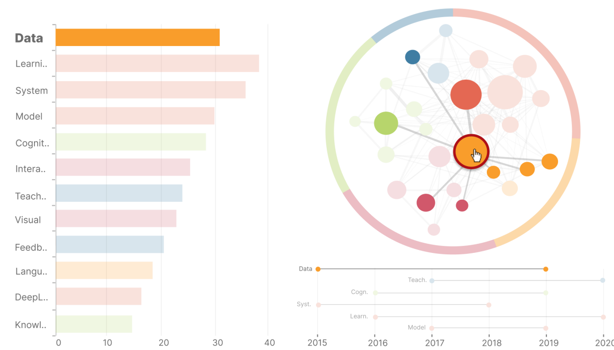


Figure 11: Inconsistent associations with the selected keyword presented in node-link diagrams and other views. Also, the bar chart and dot plot inconsistently implement keyword clipping.

consistently. The corresponding highlighting manners are missing in views other than the node-link diagram, making users lose an opportunity to gain significant insights, e.g., identifying the ranking of keywords co-presented with "Data". It is common to disregard state updates for data relationships in views without explicit connections in practice, which leaves the full advantage of multiple views untapped and even disrupts user trust-building in the MCVs.

Compatibility denotes the relationship between selections with equivalent priority, e.g., mutual exclusion and coexistence, and the integration logic that resolves them to the system state, e.g., intersection and concatenation. The inconsistent compatibility may lead to a violation of **G1**, **G2** and **G4**, which undermines the system's usability and cause failed interactions. As an easy-to-learn and user-friendly interface, *Homefinder* [35] facilitates dynamic queries with multiple data attributes by performing the intersection of selected subsets in each view consistently. In contrast, the example shown in Fig. 6 is likely to violate users' cognition. When views differ in selection modes or have inconsistent compatibility to handle multiple selections, the cognitive burden on users tends to increase.

Prioritization. The supported actions can have different priorities to change the system state. But inconsistent prioritization between views can lead to violations of **G1**, **G2** and **G4**. For definitive and tentative selection, there are four potential prioritizations. (1) High priority for tentative selection, e.g., highlight the hovering element rather than pinning elements until mouse-out. (2) High priority for definitive selection, e.g., highlighting elements when hovering and switching view contexts after pinning. (3) Equal priority, as in the case of *compatibility*. (4) Independent priority, e.g., the independent state updates and visual schemes for brushing and filtering in Fig. 6. However, when different prioritizations are mixed in MCVs, the outcome of consecutive actions may be confusing without any clue.

Users often feel frustrated with the interaction due to the vast gulf between the intentions and the consequences of actions [36]. The inconsistent action mappings and state updates in MCVs can hamper user interaction and leave them uneasily predicting the results [12].

We emphasize the consistent action translations to the system state if no additional perceptual cues are provided as suggested by **G5**.

5 Summary for Inconsistency Issues

In the previous sections, we identified the inconsistency issues in visualization practice and concretized them into five categories attached to crucial links between *User*, *VIS*, and *Data*. Here we provide a brief summary to make inconsistency better sensed.

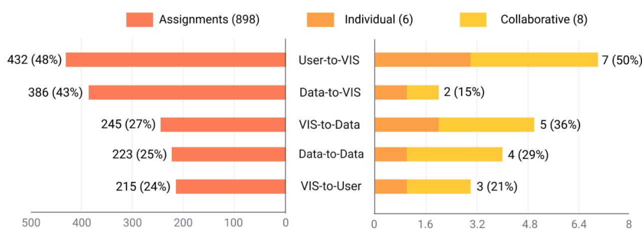


Figure 12: Statistics of inconsistency issues in the corpus [1, 2].

We counted the number of times five categories of inconsistency appeared in 898 visualizations done by novices [1] and 14 visual analytic systems completed by skilled people [2], including 6 individuals and 8 collaborative. As illustrated in Fig. 12, inconsistencies are prevalent in visualizations created by both novice and skilled people, and these inconsistency issues often occur together. Our analysis of a large corpus of multi-view visualizations found that 85.4% of them exhibited inconsistent manifestations, with 58.5% displaying more than two categories of inconsistency. These high figures indicate that consistency is not receiving enough attention in visualization practice. To address this issue, it is crucial to develop efficient guidance for preventing and detecting inconsistencies. Among the five categories of inconsistency, signal transmission (User-to-VIS) was demonstrated to be the most error-prone, with half of the cases in the corpus showing this issue. In terms of inconsistency in the visual mapping (Data-to-VIS), experienced individuals are less likely to make mistakes probably due to long-term practice, while novices are often unaware of this issue. Inconsistency issues in state update (VIS-to-Data), data transformation (Data-to-Data), and presentation mapping (VIS-to-User) occur slightly less frequently, but still affected almost a quarter of the cases and particularly in collaborative projects. This situation indicates gaps between collaborators in maintaining a consistent understanding, and the opportunity to develop theories and tools that promote collaboration consistency in visualization design and implementation. In addition, inconsistencies that arise during the coordination phase (User-to-VIS, VIS-to-Data, *Filtering* and *Sorting* of Data-to-Data) are more apparent for experienced individuals and teams, who are usually exposed to more complex multi-view systems. It suggests maintaining consistency is more challenging for MCVs with intricate coordination and dynamically updating visualizations. Therefore, existing works that focus solely on specific visual encoding in static visualizations may not be sufficient for ensuring consistency in many visual analytics systems. These findings align with the results of our preliminary study (described in Appendix A). We have made the corpus publicly available, along with the labelling of inconsistency categories [1].

6 Discussion

Consistency in visualization can facilitate comparison and inference by reducing the complexity of multiple views and easing the learning curve for users. However, it still remains a heuristic approach in practice since little teaching material or literature provides a precise definition of consistency and a thorough summary with real-life examples. As introduced in Sect. 2.3, the related works touched on visualization consistency were previously presented in sweeping terms such as "the same fields should be presented in the same way" [18] and "the way design choices are maintained in similar contexts" [15], which are difficult to understand and follow. We observed various inconsistent presentations in visualization designs and visual analytic systems. Such issues could lead to confusion or misinterpretation in user analysis, according to our preliminary study. Therefore it is significant to raise consciousness about consistency and promote efficient consistency checking in coordinated multi-view systems. To provide a pragmatic guide, we collected many interactive visualizations as a corpus [1, 2] and classified the manifestations that fail to follow the consistency guidelines. The causes of inconsistency issues are organized into each stage of visualization construction and coordination based on the model in Fig. 1, which is easy to check and locate problems. To our knowledge, current works cover only part of the consistency guidelines on visual mapping (Data-to-VIS) of static visualizations [29, 30], and neglect the discussion of interaction and coordination. Also, neither of them systematically categorized inconsistency issues nor clarified them in detail with real-life cases, which is certainly of value in practice.

To verify that our classification of inconsistency issues could serve as a useful guide for visualization practices, we invited 20 visualization stakeholders with diverse backgrounds for an evaluation. We observed that individuals who were provided with the classification and guidance manual (which is included in Appendix B) performed better on the consistency checking test. Moreover, through conducting interviews and gathering subjective feedback from the participants, we demonstrated that our research findings were effective, comprehensive, logical, easy to read, concise, and universally applicable. Further information regarding the evaluation can be found in Appendix C. In general, many visualization stakeholders can benefit from our work of consistency study:

- **T1:** Designers: Better avoid inconsistency issues when proposing visualization mock-ups and make the design of MCVs more easy-to-learn and user-friendly.
- **T2:** Developers: Improve efficiency to recognize inconsistency issues and locate code errors in application development, by reducing unsystematic and repetitive debugging activities.
- **T3:** Testers or evaluators: Accelerate their assessment for the usability of visualization from a broader perspective.

Due to length constraints, we cannot provide a comprehensive discussion of every category of inconsistency issues encountered in rich experiments. Instead, we describe their causes and consequences based on heuristic knowledge and conclusions drawn from existing works. However, this leaves ample room for future research, such as investigating how different inconsistency issues affect user perception and cognition. Nevertheless, our work represents an essential step in deepening the understanding of consistency, highlighting many research opportunities. The five stages of visualization construction and coordination susceptible to inconsistency issues reveal insufficient criteria and applications in corresponding fields. On the one hand, the visualization community still lacks sufficient knowledge of consistency in many areas, such as the relationship between consistency and other design considerations. Our work provides a corpus for MCVs research and a classification framework to support the decomposition of the complex issue. If the severity and tolerability of inconsistency can be subsequently ranked like *effectiveness* [21], it would enhance the integration of consistency with other criteria. On the other hand, our classification makes the obscure principles expressible and assessable with programming logic, supporting the creation of automated authoring tools to generate consistent multi-view dashboards [19] and visualization linters and fixers to detect inconsistency problems in MCVs [9].

7 Conclusion

This work proposes comprehensive guidelines for visualization consistency based on the summary of existing literature and practical cases. We identified five categories of inconsistency through interviews with visualization stakeholders and analyses of many visualization designs. Based on a coordination model, we clarify the possible inconsistency issues in each stage of visualization construction and coordination phases, with detailed explanations in the context of real-life examples. The step-by-step checklist we provided supports systematic and efficient checking of visualization consistency. Furthermore, our classification framework and vocabulary can deepen the understanding of consistency, raise awareness of inconsistency issues in visualization practice and facilitate the design, implementation, and evaluation of visualization with MCVs. For future work, we will explore technical solutions to automatically create consistent coordination and detect inconsistency issues in multi-view visualization based on these findings.

Acknowledgments

The authors thank anonymous reviewers for their valuable comments. This work is supported by NSFC No. 62272012 and No. 61872013.

References

[1] Gallery of visualization course assignments. [EB/OL]. `http://vis.pku.edu.cn/course/gallery/` Accessed February 23, 2023.

[2] Vis4china. [EB/OL]. `http://vis.pku.edu.cn/vis4china/` Accessed February 23, 2023.

[3] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proc. IEEE Symp. Information Visualization (InfoVis)*, pp. 111–117. IEEE, 2005.

[4] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[5] N. Boukhelifa et al. A coordination model for exploratory multiview visualization. In *Proc. Int. Conf. Coordinated and Multiple Views in Exploratory Visualization (CMV)*, pp. 76–85. IEEE, 2003.

[6] M. Brehmer et al. A multi-level typology of abstract visualization tasks. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2376–2385, 2013.

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

[8] M. Chen and A. Golan. What may visualization processes optimize? *IEEE Trans. Vis. Comput. Graph.*, 22(12):2619–2632, 2015.

[9] Q. Chen, F. Sun, X. Xu, Z. Chen, J. Wang, and N. Cao. Vizlinter: A linter and fixer framework for data visualization. *IEEE Trans. Vis. Comput. Graph.*, 2021.

[10] R. Chen, X. Shu, J. Chen, D. Weng, et al. Nebula: A coordinating grammar of graphics. *IEEE Trans. Vis. Comput. Graph.*, 2021.

[11] X. Chen, W. Zeng, Y. Lin, et al. Composition and configuration patterns in multiple-view visualizations. *IEEE Trans. Vis. Comput. Graph.*, 27(2):1514–1524, 2021.

[12] E. H.-h. Chi and J. T. Riedl. An operator interaction framework for visualization systems. In *Proc. IEEE Symp. Information Visualization (InfoVis)*, pp. 63–70. IEEE, 1998.

[13] M. C. Chuah and S. F. Roth. On the semantics of interactive visualizations. In *Proc. IEEE Symp. Information Visualization (InfoVis)*, pp. 29–36. IEEE, 1996.

[14] E. Dimara et al. What is interaction for data visualization. *IEEE Trans. Vis. Comput. Graph.*, 26(1):119–129, 2019.

[15] C. Forsell and J. Johansson. An heuristic set for evaluation in information visualization. In *Proc. Int. Conf. Advanced Visual Interfaces (AVI)*, pp. 199–206. ACM Press, 2010.

[16] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1240–1247, 2007.

[17] D. P. Jansen Y. An interaction model for visualizations beyond the desktop. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2396–2405, 2013.

[18] S. M. Kosslyn. Understanding charts and graphs. *Applied cognitive psychology*, 3(3):185–225, 1989.

[19] Y. S. Kristiansen et al. Semantic snapping for guided multi-view visualization design. *IEEE Trans. Vis. Comput. Graph.*, 2021.

[20] H. Lam. A framework of interaction costs in information visualization. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1149–1156, 2008.

[21] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graphics*, 5(2):110–141, 1986.

[22] T. Munzner. A nested model for visualization design and validation. *IEEE Trans. Vis. Comput. Graph.*, 15(6):921–928, 2009.

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

[24] C. North, N. Conklin, and V. Saini. Visualization schemas for flexible information visualization. In *Proc. IEEE Symp. Information Visualization (InfoVis)*, pp. 15–22. IEEE, 2002.

[25] C. North and B. Shneiderman. A taxonomy of multiple window coordination. Technical report, University of Maryland, 1997.

[26] C. North and B. Shneiderman. Snap-together visualization: can users construct and operate coordinated visualizations? *International Journal of Human-Computer Studies*, 53(5):715–739, 2000.

[27] T. Pattison and M. Phillips. View coordination architecture for information visualisation. In *Proc. ACM International Conference Proceeding Series (ICPS)*, vol. 16, pp. 165–169. Citeseer, 2001.

[28] P. G. Polson. The consequences of consistent and inconsistent user interfaces. *Cognitive science and its applications for human-computer interaction*, pp. 59–108, 1988.

[29] Z. Qu and J. Hullman. Evaluating visualization sets: Trade-offs between local effectiveness and global consistency. In *Proc. IEEE Workshop on Beyond Time and Errors on Novel Evaluation Methods for Visualization (BELIV)*, pp. 44–52. IEEE, 2016.

[30] Z. Qu and J. Hullman. Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring. *IEEE Trans. Vis. Comput. Graph.*, 24(1):468–477, 2017.

[31] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proc. Int. Conf. Coordinated and Multiple Views in Exploratory Visualization (CMV)*, pp. 61–71. IEEE, 2007.

[32] S. F. Roth, M. C. Chuah, S. Kerpedjiev, J. A. Kolojejchick, and P. Lucas. Toward an information visualization workspace: Combining multiple means of expression. *Hum.-Comput. Interact.*, 12(1):131–185, 1997.

[33] M. Scherr. Multiple and coordinated views in information visualization. *Trends in information visualization*, 38:1–33, 2008.

[34] L. Shao et al. Modeling layout design for multiple-view visualization via bayesian. *Journal of Visualization*, 24(6):1237–1252, 2021.

[35] B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, 1994.

[36] E. Shove. *The design of everyday life*. Berg, 2007.

[37] M. Sun, A. R. Shaikh, H. Alhoori, and J. Zhao. Sightbi: Exploring cross-view data relationships with biclusters. *IEEE Trans. Vis. Comput. Graph.*, 28(1):54–64, 2022.

[38] M. Tory and T. Moller. Rethinking visualization: A high-level taxonomy. In *Proc. IEEE Symp. Information Visualization (InfoVis)*, pp. 151–158. IEEE, 2004.

[39] J. J. Van Wijk. The value of visualization. In *Proc. IEEE Visualization (VIS)*, pp. 79–86. IEEE, 2005.

[40] E. Wall, M. Agnihotri, L. Matzen, K. Divis, M. Haass, A. Endert, and J. Stasko. A heuristic approach to value-driven evaluation of visualizations. *IEEE Trans. Vis. Comput. Graph.*, 25(1):491–500, 2018.

[41] J. Walny, C. Frisson, M. West, D. Kosminsky, S. Knudsen, S. Carpendale, and W. Willett. Data changes everything: Challenges and opportunities in data visualization design handoff. *IEEE Trans. Vis. Comput. Graph.*, 26(1):12–22, 2019.

[42] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proc. Int. Conf. Advanced Visual Interfaces (AVI)*, pp. 110–119. ACM Press, 2000.

[43] M. O. Ward, G. Grinstein, and D. Keim. *Interactive data visualization: foundations, techniques, and applications*. CRC press, 2010.

[44] M. O. Ward and J. Yang. Interaction spaces in data and information visualization. In *VisSym*, pp. 137–145, 2004.

[45] C. Weaver. Building highly-coordinated visualizations in improvise. In *Proc. IEEE Symp. Information Visualization*, pp. 159–166, 2004.

[46] C. Weaver. Visualizing coordination in situ. In *Proc. IEEE Symp. Information Visualization (InfoVis)*, pp. 165–172. IEEE, 2005.

[47] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1129–1136, 2007.

[48] Y. Wu, R. Chang, J. M. Hellerstein, A. Satyanarayan, and E. Wu. Diel: Interactive visualization beyond the here and now. *IEEE transactions on visualization and computer graphics*, 28(1):737–746, 2021.