DMiner: Dashboard Design Mining and Recommendation

Yanna Lin, Haotian Li, Aoyu Wu, Yong Wang, and Huamin Qu

Abstract—Dashboards, which comprise multiple views on a single display, help analyze and communicate multiple perspectives of data simultaneously. However, creating effective and elegant dashboards is challenging since it requires careful and logical arrangement and coordination of multiple visualizations. To solve the problem, we propose a data-driven approach for mining design rules from dashboards and automating dashboard organization. Specifically, we focus on two prominent aspects of the organization: *arrangement*, which describes the position, size, and layout of each view in the display space; and *coordination*, which indicates the interaction between pairwise views. We build a new dataset containing 854 dashboards crawled online, and develop feature engineering methods for describing the single views and view-wise relationships in terms of data, encoding, layout, and interactions. Further, we identify design rules among those features and develop a recommender for dashboard design. We demonstrate the usefulness of DMiner through an expert study and a user study. The expert study shows that our extracted design rules are reasonable and conform to the design practice of experts. Moreover, a comparative user study shows that our recommender could help automate dashboard organization and reach human-level performance. In summary, our work offers a promising starting point for design mining visualizations to build recommenders.

Index Terms—Design Mining, Visualization Recommendation, Multiple-view Visualization, Dashboards

✦

1 INTRODUCTION

MULTIPLE-VIEW (MV) visualizations composite multiple visualizations into a single cohesive representation. Due to its power to support users in exploring several perspectives of data simultaneously, a large number of MV visualizations have been created and shared on the web by various domains, from biomolecular to multimedia to business. Creating an MV visualization typically starts with selecting views of interest, followed by presenting the selected views and adding interactions between views [1]. This paper refers to the presentation and interactions as layout arrangements and coordination, respectively. Arranging and coordinating views are vital when explaining the widest range of usability problems in visualizations [2]. Specifically, a proper arrangement can maximize the utility of the limited display space and improve the effectiveness and expressiveness of the information exchange, improving the usability of the system [3]. Besides, coordination among visualizations can make cross-view data relationships more apparent and reduce users' cognitive burden [4].

However, it remains challenging to create effective MV visualizations with a proper view arrangement and coordination. From a theoretical perspective, existing guidelines on MV visualization designs focus on high-level recommendations, e.g., drawing users' attention to the right view and making cross-view data relationships more obvious [1]. They are insufficient in providing lay users with actionable

suggestions to adjust their designs. From a practical perspective, while researchers have recently developed some recommenders or authoring tools to assist lay users in creating visualizations and MV visualizations, few of them have focused on the arrangement and coordination among views. Existing tools (e.g., Tableau [5], Power BI [6], and MultiVision [7]) provide default layout templates that require manual adjustment to achieve satisfactory MV visualization designs. This process is tedious and time-consuming, given that the potential layouts increase exponentially with the increasing number of views.

We present DMiner, a data-driven framework for mining dashboard design and automating the layout arrangement and view coordination for MV dashboards (Figure 1), thus reducing the design burden of designers. Specifically, MV dashboards are one of the most common genres of MV visualizations [8]. In this paper, we use the term MV dashboards and dashboards interchangeably to represent multiple-view dashboards. Given that end-to-end ML-based visualization recommenders suffer from poor explainability and can confuse end-users [9], we aim to develop an explainable approach by first mining design rules from an MV dashboard dataset and further recommend appropriate layout arrangement and view coordination in MV dashboards.

Due to the lack of MV dashboard datasets, we first crawled a large number of dashboards created by Tableau, a common dashboard authoring tool, from GitHub [10]. We deduce a set of features that influence the arrangement and coordination of these views through reviewing prior studies, and further identify the mappings among them (Figure 1 (A)). With the collected dataset, we then extract features from two perspectives: 1) the single-view features describing each view in terms of its visual encodings (e.g., color), encoded data (e.g., data types), and

- Y. Lin, H. Li, A. Wu, H. Qu are with the Hong Kong University of Science and Technology.
  E-mail: {ylindg, haotian.li, awuac}@connect.ust.hk, huamin@cse.ust.hk

- Y. Wang is with the Singapore Management University.
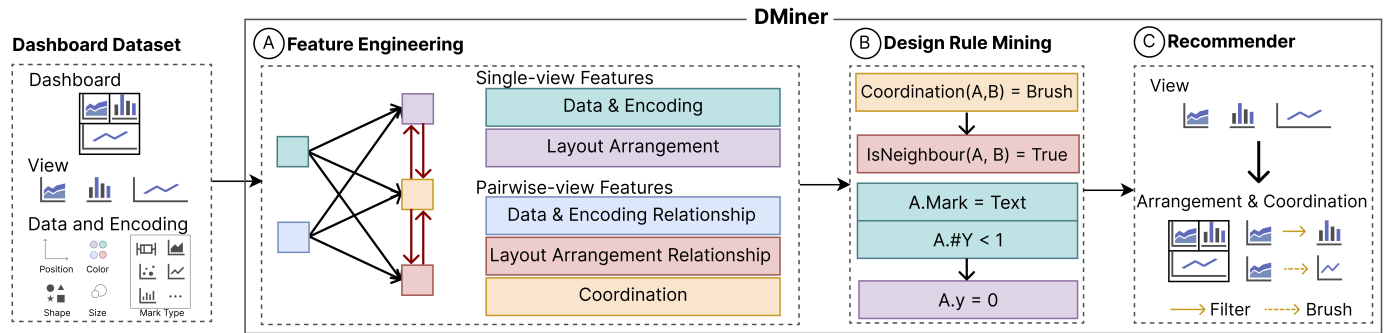 E-mail: yongwang@smu.edu.sg

Figure 1. The workflow of DMiner. This paper proposes DMiner as a framework for dashboard design mining and automatic recommendation. With the dashboard dataset as the input, DMiner: (A) first surveys a set of features important for dashboard design, and then extracts those features to delineate dashboard designs comprehensively. These are categorized into two types, i.e., single-view features such as data and encoding and pairwise-view features such as coordination and relative position; (B) then mines design rules using decision rule approach, and further filters them; and (C) finally leverages these rules for recommending dashboard arrangement and coordination.

its layout (e.g., position); and 2) the pairwise-view features describing pairwise relationships between views, including the data relationship (e.g., overlapping data columns), encoding relationships (e.g., same encoding), spatial relationships (e.g., relative angle), and coordination (e.g., filter).

Building upon those features, we mine the mappings using the decision rule and filter them to distill the design rules of dashboards (Figure 1 (B)). Specifically, based on the data and encodings of every single view and the data relationship and encoding relationship between views, the rules infer the layout of each view and the spatial and coordination relationship between view pairs. For instance, one design rule in Figure 1 (B) shows that a text visualization ("A.Mark = Text") without data encoding on the Y-axis ("A.#Y < 1") tends to place at the top ("A.Y = 1"). Using the extracted rules, we develop a recommender that recommends the optimal arrangement and coordination by ranking the obedience of design rules (Figure 1 (C)).

We demonstrate the usefulness and effectiveness of DMiner through an expert study and a comparative user study using the collected Tableau dashboard dataset. We invite four experts to rate the appropriateness of the 10% of extracted rules, and showcase those rules with higher expert scores. Also, we compare our recommender with the existing tool, the crawled dashboard, and experienced designers. We believe that our work can be a starting point for automating the layout arrangement and view coordination in MV dashboards. Our main contributions are as follows:

- DMiner, a novel data-driven framework for dashboard design mining and recommendation.
- An expert study to demonstrate the appropriateness of extracted rules and a comparative study to show the effectiveness of the recommender.
- A dataset of 854 real-world dashboards with detailed information such as metadata, encodings, and coordination, which can benefit future research on dashboards.

2 RELATED WORK

Our research is related to prior studies on multiple-view visualizations, visualization recommendations, and design knowledge mining.

2.1 Multiple-view Visualization

Multiple-view visualizations (MVs) have gained extensive research interest in the visualization community. A large body of research aims to advance the theoretical underpinnings of MVs through empirical methods. Baldonado et al. [1] drew from a workshop to present eight guidelines for using MVs in information visualization along three dimensions: the selection, presentation, and interaction of views. Roberts et al. [11] contributed a state-of-the-art report highlighting seven fundamental research areas of MVs: data processing, view generation, exploration techniques, coordination, tool infrastructure, human interface, usability and perception. Sarikaya et al. [8] summarized the design space of dashboards by analyzing the intentions of a chosen corpus of 83 dashboards. Followed by this work, Bach et al. [12] gathered 36 more dashboards and proposed a more fine-grained design space focusing on the dashboard structure, visual design, and interactivity.

Researchers have also contributed several experiments to understand the design of MVs. For instance, Qu and Hullman [13] studied how visualization authors consider the importance of encoding consistency between views when designing MVs. Langner et al. [14] investigated how users interacted with MVs on large screens and concluded with design suggestions. While those studies contributed valuable design knowledge, they rely on large-scale experiments and require considerable human effort. We take a different perspective by automatically extracting MV design knowledge from a larger scale corpus.

One of the key challenges in designing MVs is to relate data relationships among views [15]. Coordination techniques such as brushing and linking can reveal cross-view data relationships and have been widely used in visualization tools (e.g., Tableau [5], Power BI [6], Jigsaw [16]). Despite being useful, they require users to trigger interaction events (e.g., brushing) and pay adequate attention to subsequent changes on other views. Thus, researchers have advocated the use of proximity-based methods that spatially organize relationship components [17]. Inspired by those studies, we aim to automatically recommend coordination and spatial arrangement of views to facilitate the design of MVs. We focus on MV dashboards, which are one of the most popular genres of multiple-view visualizations [8].

Through a review of existing work, we develop a set of features that influence the arrangement and coordination of MV dashboards, extract design rules from a corpus of Tableau dashboards and further recommend appropriate arrangement and coordination of multiple views.

2.2 Visualization Recommendation

Authoring effective and elegant visualizations is a challenging task even for professionals, since it requires the consideration of many aspects such as data insights, perceptual effectiveness, and aesthetics. Researchers have proposed many visualization recommendation systems to assist in data analysis. Several recommenders (e.g., Data2Vis [18], VizML [19], Draco [20], and KG4Vis [21]) focus on recommending visual encodings of a single visualization.

However, a single visualization is often insufficient in supporting in-depth data analysis, as it often requires progressive and iterative exploration of different data subsets [22]. Thus, researchers have started to investigate the problem of recommending multiple visualizations. For instance, VizDeck [23] utilizes users' preferences to select data for presentation and organize dashboards. Voder [24] supports the interactive exploration of data facts associated with charts and natural language descriptions. To further reduce manual effort, Tundo et al. [25] allowed users to select dashboard templates to transform the declarative definition they create into dashboards. MultiVision [7] and Dashbot [26] recommend analytical dashboards given an input dataset in an end-to-end manner. However, those systems are either template-based and do not consider the underlying data, or do not recommend layout arrangement and view coordination, which requires considerable human effort. Our work fills this gap by developing a recommender for arranging views in an MV dashboard.

Research in the field of data storytelling has studied approaches for composing multiple visualizations in a logical manner. For instance, researchers have proposed methods to arrange visualizations in a logical sequence to enhance storytelling [27], [28]. DataShot [29] and Calliope [30] group multiple visualizations into a coherent topic according to insights derived from the visualizations. Different to them, we study the layout arrangement and view coordination in MV dashboards for data analysis and comprehension.

2.3 Design Mining Visualizations

There are an increasing number of multiple-view visualizations created and shared by people from different domains, which has inspired many researchers to mine and extract multiple-view visualization design knowledge from them. Some researchers have invested effort into adopting the statistical method to mine the visualization usage. Al-maneea and Roberts [31] collected MVs from the published papers, and answered what the most common number of views is and what the popular tiles are by statistically counting their manual-label chart type and layouts. Beyond counting, Chen et al. [32] further leveraged some statistical methods like condition probability to perform the configuration and composition analysis, and integrated the findings into a system for exploration and recommendation. Following this work, Shao et al. [33] employed Bayesian probabilistic

inference to analyze the effects of design factors on layouts of MVs, and discovered some insightful layout design patterns, e.g., views for exploration with a more scattered area ratio. Though inspiring, these analyses and findings were limited to arrangements and view types, since the MVs collected in image format prevented them from accessing the underlying data and the coordination among views. Recently, Lu et al. [34] detected the position and underlying semantics (such as the numbers 1-9) of components in infographics and explored how different components are linked. Inspired by this work, we aim to mine the mappings from the characteristic of individual views and view pairs to the arrangement and coordination in an MV dashboard.

3 DASHBOARD DATASET

Our study aims to mine design rules from existing dashboards and further automatically recommend optimal MV dashboards. However, there is a lack of off-the-shelf dashboard datasets that can enable automated MV dashboard design. Specifically, existing datasets of MV dashboards [31], [32], [33] contain only the information of individual views, such as position, size, and chart type. They do not provide the metadata (e.g., the data operations) and the interactions between views, which are necessary information for developing visualization and dashboard recommenders [7], [19]. To address this problem, we build a new dataset from the Internet. The detailed procedures for constructing the dashboard datasets will be introduced as follows:

Data Crawling. We first crawl Tableau dashboards from GitHub, i.e., searching and downloading Tableau workbooks with *.twbx* extensions. There are multiple advantages to using Tableau dashboards. First, Tableau workbooks contain all the necessary information to render the corresponding dashboards (e.g., the underlying metadata, visual encodings, and coordination among views). Second, the information is stored in a structured XML format, which can be processed automatically at scale and can avoid the heavy manual labelling procedures required in previous studies [31], [32], [33]. Finally, Tableau is one of the most popular tools for creating dashboards, and its workbooks are widely shared on GitHub. Thus, the collected dataset is diverse regarding its creators, design themes and styles, offering opportunities to mine common design rules.

Automatic Data Processing and Cleaning. The datasets crawled from GitHub suffer from noise, affecting our analysis results. To improve the dataset quality, we first perform data cleaning by parsing and filtering the raw workbooks (.twbx). As shown in Figure 2, we parse the workbooks to derive information including the metadata (e.g., data types and data operations), encodings for each view, the coordination among views, and the arrangement of each view in a dashboard. Subsequently, we remove workbooks without dashboards (e.g., the workbooks containing only individual views). Dashboards using multiple data sources are also abandoned since the lack of relational schema between data sources hinders data processing. Finally, we discard dashboards without any view coordination (e.g., cross-view interactions), since we aim to mine and recommend view coordination. According to Tableau, there are two coordination types, namely filtering and brushing. Specifically, *filtering*
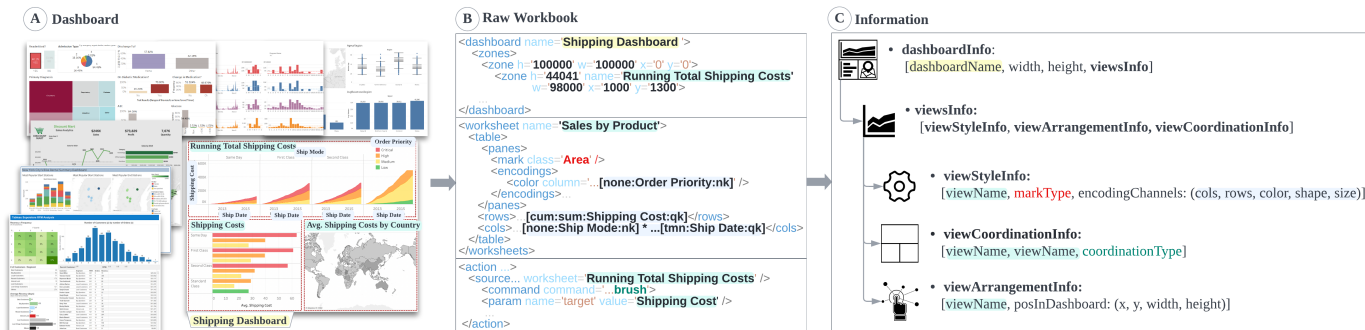
Figure 2. Overview of the data collection pipeline: (A) We crawled a diverse set of Tableau dashboards from GitHub; (B) We automatically parsed the raw workbooks and (C) extracted information about the view arrangement and coordination.
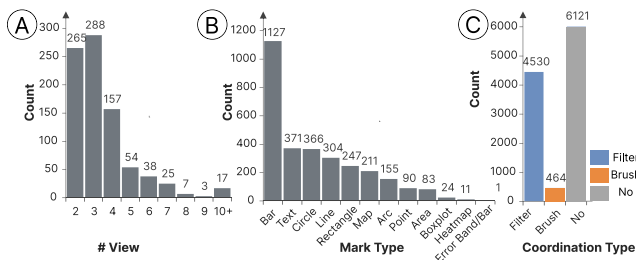

Figure 3. The basic statistics of our dashboard collection. (A) and (B) show the distribution of the number of views in each dashboard and the distribution of the mark type of each view, respectively. (C) describes the distribution of coordination types for all 854 dashboards.

removes irrelevant data objects while *brushing* highlights the related visual elements and keeps the context [35].

Manual Data Processing and Cleaning. Tableau workbooks offer a feature that automatically decides the mark type. Therefore, about half of the visualizations have an "automatic" mark type. To address this problem, we manually label the mark types according to both the definition of marks in Tableau and Vega-Lite [36], a widely-used grammar of visualizations. A detailed list of the mark types and corresponding examples is available in the supplementary material. To ensure the quality of labels, each view is labeled by two co-authors. The co-authors first finish the labelling individually, resulting in 99% agreement. Conflicts are then discussed until reaching a consensus. We remove dashboards containing views with multiple mark types, e.g., the network that includes the *circle* and *line*.

Results. We finally collected 854 dashboards with 2990 views in total for further exploration, and more details can be found in the supplementary material. Specifically, Figure 3 shows the basic information of the dashboards, where (A), (B), and (C) represent the distribution of the number of views of each dashboard, the mark type of each view, and the coordination among view pairs, respectively. Figure 3 (A) and (B) show that the distribution of the number of views aligns well with the observations from previous studies [31], [32]. For exmaple, 2- and 3-view dashboards are the most common, and *bar* is the most popular mark type. Figure 3 (C) shows that half of the view pairs possess coordination, which is almost *filtering*.

4 DMINER

DMiner aims to automatically extract design rules for MV dashboards from existing dashboard datasets and guide the subsequent design of MV dashboards, which is common in visualization recommenders (e.g., Draco [20]). Figure 1 provides an overview of DMiner. We first survey previous relevant studies to identify a set of key features that indicate important considerations for MV dashboard design. We then extract those features from our dataset and mine design rules. Finally, we develop a recommender for MV dashboard design based on the extracted rules. The detailed procedures of feature engineering, rule mining, and recommender will be introduced in the following subsections.

4.1 Feature Engineering

Figure 4 provides an overview of the extracted features and their mappings. Unlike existing works that propose single-view features (e.g., the absolute position of views), we further introduce pair-wise features (e.g., the relative position of views), which allows us to delineate the arrangement and coordination between views in a more fine-grained manner.

Our design choices for feature selection and the mappings among them are built upon prior theoretical research about dashboards and multiple visualizations [32], [37], [38], [39], [40]. Specifically, they have suggested that the data and encoding features of views are crucial for arranging and coordinating the view into a cohesive dashboard. Therefore, we choose those mappings, which are denoted as black arrows in Figure 4. The bottom part of Figure 1(B) shows an example mapping from *data and encoding features* of a view to its *arrangement features*, which describes that a text view (i.e., $A.Mark = Text$) without data encoding on Y-axis (i.e., $A.\#Y < 1$) tends to be placed at the top (i.e., $A.y = 0$). Also, prior studies have shown that the arrangement and coordination of a dashboard are interrelated [31], [33]. Therefore, we have further added such mappings in DMiner, as indicated by the red arrows in Figure 4. For example, the upper part of Figure 1(B) shows that two views, with one brushed by the other (*coordination features*), should be put closely (*arrangement relationship features*). In summary, we have defined ten mappings among the features describing dashboard designs. Figure 5 shows the formal definitions of all the extracted features. They can be divided into two major groups, i.e., single-view features and pairwise-view features. Single-view features include features about
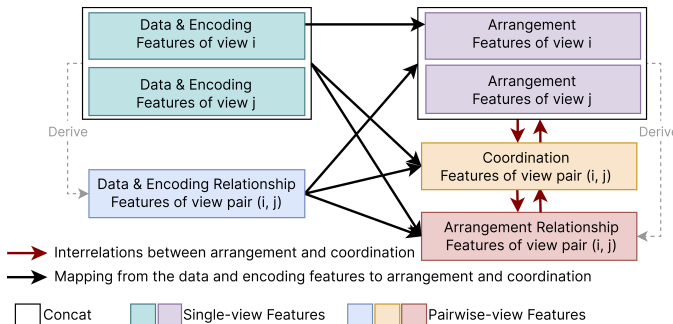
Figure 4. The targeted mappings in our design rule mining. We identified ten mappings in black and red between features in designing dashboards, while the gray dotted lines show the derivation relationship. These black mappings focus on how to utilize data and encoding information of single views for arranging these views in a dashboard and adding coordination among them. The red mappings between arrangement features and coordination features are used to further comprehensively describe the dashboard design.

the data, encoding, and arrangement of each view in a dashboard. Pairwise-view features are extracted to represent the relationship between two views. Specifically, we extract the data relationship, encoding relationship, arrangement relationship, and coordination relationship between two views. In summary, there are 33 single-view features and 41 pairwise-view features. A detailed feature list is available in the supplementary material.

4.1.1 Single-view Features

Single-view features are extracted to describe each individual view in the dashboard. These features can be roughly categorized into two types.

Data and encoding features can reflect what data is used in the view and how the data is visually encoded, which has been considered essential for view arrangements in a previous study [32]. Inspired by previous studies that utilize features to characterize single-view visualizations [19], [21], we first extract data features regarding data types (i.e., numerical, nominal, and ordinal) and data operations (e.g., count and sum). Since the numbers of data fields in different views are inconsistent, it is infeasible to use each field's data type and operation as features. Instead, as shown in Figure 5, we define an operator, i.e., *CountFunc*, to compute the frequency of different data types and data operations as features of views. For example, the bar chart in Figure 3 (A) uses one ordinal field on X-axis and one numerical field on Y-axis. Regarding encoding information, we extract features about the mark type of the view and the usage of different encoding channels (i.e., position, size, color, shape). Similar to data features, we also summarize the usage of each channel, for example, how many data fields are encoded using the position channel on X-axis.

Layout arrangement features describe how the view is placed in the dashboard. We propose a series of arrangement features to delineate the position and size information of each view. The display area of a dashboard is evenly divided into $n \times n$ grids ($n > 0$), and we assume that each view takes a rectangle area and does not overlap with each other. In this paper, n is set to 4 to strike a balance between performance and efficiency in our subsequent evaluations.

However, our mining algorithm can also work well when n is other values. As a result, besides the features delineating a view's arrangement with its original position and size, the arrangement features using the grids are also extracted. Given that each dashboard has a different size, the size of all dashboards is normalized to 4*4 to enable comparisons. That is, the width and height of each dashboard is 4, and the width and height of each view is one of the values $[1, 2, 3, 4]$.

4.1.2 Pairwise-view Features

Previously we introduced how we extract the single-view features to characterize each individual view in the dashboard. To further reveal the relationships across different views in the dashboard, we introduce a group of pairwise-view features. In our study, we propose three types of pairwise-view features:

Data and encoding relationship features are computed via aggregation functions on two views. Similarly, due to the inconsistent number of fields among views, it is infeasible to calculate the statistical values of two views like correlation [19]. We propose two main types of aggregation operators shown in Figure 5. The first type is binary operators to compare the number of data fields, e.g., *IsEqual* determines whether two views encode the same number of fields for a given encoding channel. The second type is set operators concerning the overlapping between data encodings, e.g., *IsOverlapping* and *CountOverlapping* decide whether and how many data fields encoded by two views are overlapping.

Layout arrangement relationship features concern the relationships of both the sizes and relative layouts of two views. For the former, we similarly apply aggregation to compare the sizes between two views, e.g., *IsLarger* compares whether the size of one view is larger than the other. For the latter, we compute the relative angle and distances between two views. We further introduce *IsNeighbour* to decide whether two views are adjacent.

Coordination features describe the interactions between two views. As mentioned in Section 3, there are two types of coordination in Tableau, namely brushing and filtering. Specifically, *filtering* removes irrelevant data objects while *brushing* highlights the related visual elements and keeps the context [35].

4.2 Design Rule Mining

The goal of DMiner is to find effective arrangements and coordination for the multiple views of a dashboard. Thus, we aim to model the mapping from data and encoding features to arrangement and coordination as well as the interrelations between arrangement and coordination within multiple dashboard views, as shown in Figure 4. Here, we adopt the decision rule approach [41] to achieve this goal. The reasons for choosing the decision rule approach are as follows: 1) it allows us to investigate the mutual influences among those features in a data-driven manner; 2) it embraces algorithmic explainability and is easy for humans to interpret. Specifically, given a set of dashboard features $\mathcal{F}$ as detailed in Figure 5, we aim to mine rules that imply $X \Rightarrow Y$, where $X, Y \subseteq \mathcal{F}$. All these rules are automatically learned and extracted from the collected dashboard dataset.

A Single-View Feature

SingleViewFeature := DataEncoding | LayoutArrangement

DataEncoding := CountFunc(DataEncodingAttribute)
DataEncodingAttribute := DataType | DataAggregation | Encoding
DataType := numerical, nominal, ordinal
DataAggregation := count | sum | ...
Encoding := position | size | color | shape

LayoutArrangement := Position | Size | GridOccupancy
Position := X | Y, Size := Height | Width | Area
X/Y/Height/Width := <Number>
GridOccupancy:= <Matrix>

Colored cells are marked as 1 and remainings are 0.

B Pairwise-View Feature

PairwiseViewFeature := DataEncodingRelationship | LayoutRelationship | Coordination

DataEncodingRelationship := AggregationFunc(View1, View2, DataEncodingAttribute)
AggregationFunc := IsEqual | IsLess | IsMore | IsOverlapping | ...
OverlapCalcFun := IsOverlapping | CountOverlapping

IsEqual (View1, View2, color)
% whether two views encode the same attribute for the color channel

LayoutRelationship := SizeRelationship | RelativeLayout
SizeRelationship := AggregationFunc(View1, View2, SizeAtttribute)
RelativeLayout := Angle | Distance | IsNeighbour
Angle/Distance := <Number>

IsLarger (View1, View2, area)
% whether view1 is larger than view2 in terms of area

Coordination := brush | filter | none

Notation: A := X | Y means that A is either X or Y.
Note: The listed features are either <Boolean> or <Number> or <Nominal> values.
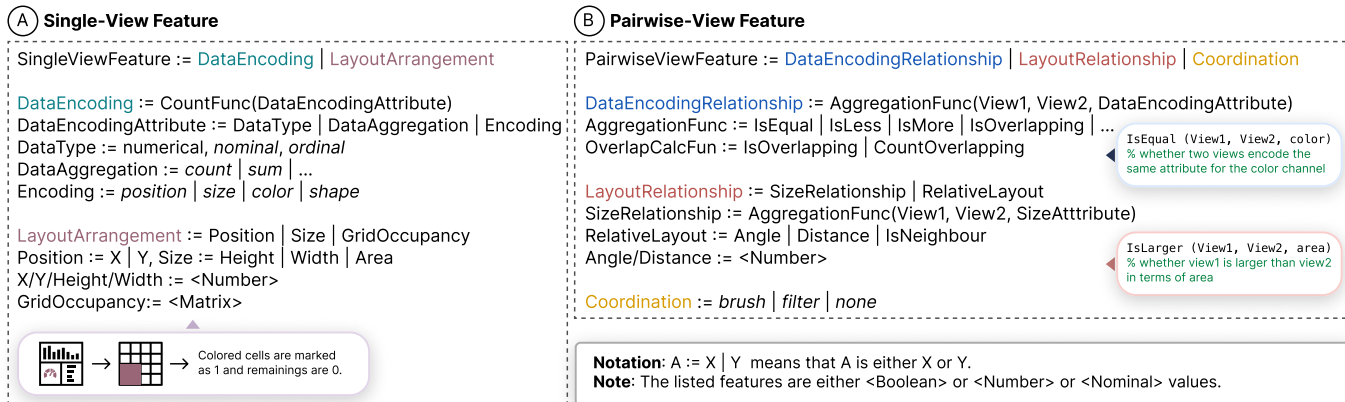
Figure 5. The formal definition and representation of extracted features. We classify features into single-view and pairwise-view features: (A) The former describes data, encoding, and layout of a single view. (B) The latter describes relationships of data, encoding, and layout between two views as well as the coordination.

Model Selection. As mentioned above, we have formulated the problem as mining decision rules. To solve the problem, we chose one of the decision rule algorithms, RuleFit Binary Classifier [42]. It is efficient since it produces a set of unordered independent rules, which can be checked in parallel rather than in series [43]. The extracted rules of RuleFit follow the structure: $condition \rightarrow target$ with each rule's coefficient and importance. The coefficient and importance of a rule describe how and how much the *condition* contributes to the *target*. Specifically, if the *condition* is obeyed, then the corresponding *target* tends to be true for positive coefficients and false for negative coefficients.

Feature Processing. Since the *condition* and the *target* of rules extracted by RuleFit have to be Boolean variables, we need to process the extracted features before feeding them into the model. Specifically, we convert numerical and categorical features into Boolean variables by setting thresholds or asserting equivalence. To handle numerical values, we set manual thresholds by splitting the interval by the average value of two-edged cut-offs (i.e., larger or smaller than the mean). The purpose of two-edged cut-offs is to improve interpretability. It can be challenging for humans to understand more intervals, e.g., *the distance of two views smaller than 0.5* is easier for sense-making than *the distance either within* $(0.0, 0.15]$ *or* $(0.3, 0.45]$. The final derived Boolean features are represented using binary vectors [44].

Training and Post-processing. We train a RuleFit binary classifier for each *target* feature of each mapping identified in Figure 4. In total, we train 208 models. Specifically, to increase the interpretability of the rules, we reduce the condition complexity of each rule by limiting each *condition* with at most 2 features [45]. We randomly partition dashboards into a 75% (640) training set and a 25% (214) test set. The trained models achieve an average of 73% accuracy on the training set and 71% accuracy on the test set. Considering the huge amount of models, we select the top 3 rules with positive coefficients and the highest importance for each model to avoid overfitting issues, resulting in 624 decision rules.

4.3 Recommender

Given the design rules extracted in Section 4.2, the recommender of DMiner (Figure 1 (C)) can further automatically recommend appropriate view arrangement coordination for the views of a dashboard. Figure 6 provides an overview of the recommender pipeline. Similar to prior optimization-based visualization recommendation approaches [46], our recommender also enumerates all the possible designs and recommends the designs with the highest scores. Specifically, it first searches all possible arrangements (i.e., positions and sizes for each view) to generate candidates. Then, it computes the score for each candidate by checking whether it obeys or violates the extracted decision rules. Inspired by [20], the cost score is a weighted sum of violated decision rules, since different rules make different contributions to the final designs. We use the importance of each decision rule as the weight, which measures how the rule is important to the prediction introduced in Section 4.2.

However, it is time-consuming and infeasible to calculate scores for all candidates. The number of possible candidates is huge, since it grows exponentially with the number of views. To reduce the computational complexity, we adopt the common strategy in previous studies on visualization recommendations (e.g., SeeDB [47] and QuickInsights [48]), i.e., pruning, to balance efficiency and performance. We first consider only the rules that map **S**ingle-view data and encoding features to **S**ingle-view layout arrangement features (denoted as **S2S** rules), since it only considers single view and has smaller complexity than pairwise-view computations. The top 1% recommendations with the highest average score are subsequently fed into the next step, i.e., to compute scores based on all other rules. The percentage 1% is decided on an empirical basis to reduce the running time to seconds, which is desirable in real-world scenarios. Finally, we recommend the arrangement and coordination with the highest score.

5 EVALUATION

To evaluate DMiner, we conducted an expert study to evaluate the appropriateness of our extracted design rules
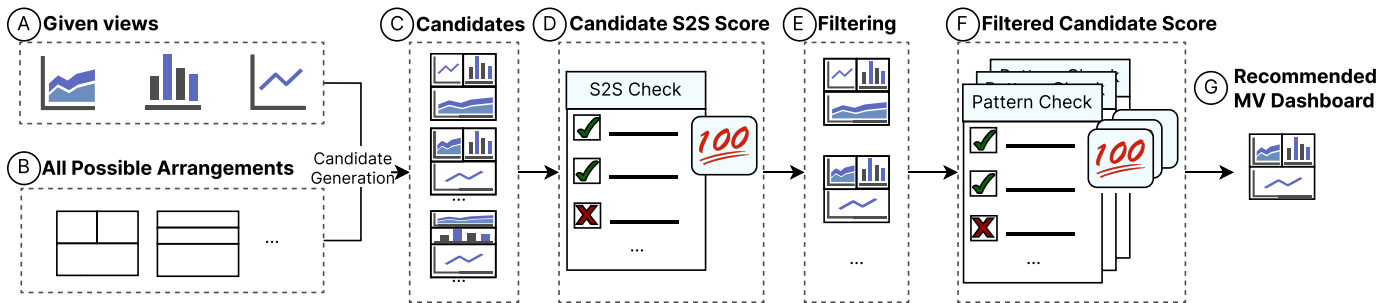
Figure 6. The recommender framework. Given several views and pre-computed possible arrangements, we first generate the candidate dashboards. Then we apply S2S rules to prune unreasonable candidates to improve efficiency. The remaining candidates are further assigned scores and ranked using other rules. Finally, we recommend the candidate with the highest score.

(Section 5.1), as well as a user study to demonstrate the effectiveness of the recommender (Section 5.2).

5.1 Expert Study

We conducted a study with four dashboard design experts to gather their quantitative and qualitative feedback about the appropriateness of the extracted design rules. We presented the design rules extracted by our approach to the experts, and asked them to indicate the extent to which the extracted rules match their expert knowledge of dashboard designs. To demonstrate the effectiveness of our approach, we discussed the top 5 rules with the highest appropriateness scores in detail (Table 1) and summarized the feedback from the experts.

5.1.1 Study Setup

Design Rules. It is time-consuming to ask our experts to evaluate all the extracted rules. Thus, we selected Top 10% (62) rules with the highest accuracy on the test dataset. To increase the readability, we translated rules into natural language sentences. For example, the rule in Figure 7 (1) was translated into "**If** *View A is Text,* **and** *it does not have fields on Y-axis,* **then** *View A should be of the height of 1*". Moreover, considering that some rules may present similar conditions or semantic meanings, we further organized them into 22 *processed* rules. For example, the second and the third rule in Table 1 were merged from several rules with the same *if* conditions. These 22 processed rules with the associated original rules were presented for the experts to evaluate and justify their validity.

Participants. We invited 4 experienced visualization or dashboard design experts (3 males, with age 31.25 ± 2.87). **Expert 1** (**E1** for short), a full professor at a local university, has engaged in visualization analytics for ten years, and has been working on MV dashboard design for the past few years. **E2** has worked as a research scientist in a company for nine years, specializing in using dashboards for text analytics and business analysis. **E3**, a research assistant professor, has been designing dashboards for spatial-temporal data analytics for the past seven years. **E4**, a strategy analyst in a finance company for 3.5 years, needs to analyze trading data daily using dashboards. None of them is color-blind.

Procedure. Each expert study lasted about 1.5 hours, and we sought the experts' consent to record the entire process. The studies were preceded by a 10-minute introduction

to our work, including the goals and the corresponding features in Section 4.1. In the expert studies, each expert was presented with 22 processed rules with 62 original rules. Also, we provided a user interface for experts to explore some supporting example dashboards for each original rule. Experts were asked to score the appropriateness of 22 processed rules on a 7-point Likert scale from 1 (the least reasonable) to 7 (the most reasonable) and give qualitative feedback about their scores. All the experts were encouraged to think aloud throughout the process.

5.1.2 Result Analysis

Our extracted design rules were appreciated by the experts. Considering that each processed rule represented a different number of original rules, we got a weighted average score of 4.44 ± 1.32, with the number of original rules as the weight. According to the qualitative feedback from experts, our extracted design rules were reasonable and aligned well with their design knowledge, thus reducing the burden of dashboard design. **E4** commented that our rules used *low-level and more specific features*, and were therefore easier to understand and follow. In this subsection, we present the top five rules with the highest average scores (Table 1), and summarize the feedback from the experts. An example case is used to illustrate four of these rules, as shown in Figure 7.

Table 1
This table shows the top 5 rules that received the highest score from the experts.

No.	Rule	Score
1	**If** View A is *Text* (not *Text* table), **then** View A should be of the height or width of 1.	6.5
2	**If** View A and View B are of the same chart type, **and** they use the same fields on Y-axis, **then** View A should be to the left or right of View B.	6.5
3	**If** View A and View B share more than 50% of the same fields, **and** they use color for the same fields, **then** View A should brush View B or View B should brush View A.	6.25
4	**If** View A is *Text* (not *Text* table), **then** View A should be on the top right, top left, or top of other view types.	6
5	**If** View A has more fields than View B, **then** View A should be on the bottom right, bottom left, or bottom of View B.	5.5

Adjust the size of the view according to the importance of the view. The first rule receiving a score of 6.5 from experts is that the *Text* view (not the *Text* table) tends to

Figure 7. An example dashboard to illustrate identified design rules. The dashboard consists of four views, (A) - (D). Part of the data and the information of each view in the dashboard are shown in two tables, (E) and (F), respectively. In the dashboard, the application of four design rules ((1) - (4)) has been observed. For example, (1) represents that the height of *Text* view in (A) tends to be 1, and (3) describes that this *Text* view in (A) tends to be placed on top of other views.

have the smallest width or height, such as Figure 7 (A) and (1). **E2** and **E3** confirmed that "*Text view is usually used as an assistive view to describe the data analysis purpose and does not have too many meanings. Thus, it is often equipped with either the smallest height or the smallest width*". **E1** mentioned that "*This reminds me that the size of the view should be related to the importance of the view, especially given the limited space available in the dashboard*".

Configure the two views used for comparison in a similar and close manner. The second rule is that, if two views possess the same mark type and use the same fields on the Y-axis, they are preferred to be arranged side by side ($score = 6.5$). All the experts inferred that these two views were for comparison, and recognized that they preferred to arrange views horizontally for comparison. **E4** gave this rule a score of 7 and commented, "*Two views with the same chart type and Y-axis must be in the same information hierarchy in a dashboard. I would not hesitate to arrange them side by side*". Both **E2** and **E4** said that the views for comparison should be configured "*as similar as possible, like the same width, the same height, and the same scale*". In this way, users can compare them effectively. While **E3** mentioned, "*In some cases, I might also arrange two comparison views vertically, if there are also other views that need to be arranged horizontally. However, most of the time, I prefer the horizontal orientation*". **E1** pointed out, "*This condition reminds me of the concept of 'small multiples' (proposed by Tufte [37]), which describes the practice of arranging two views of the same visual type together to facilitate comparison*".

Add some necessary coordination to facilitate data exploration. The third rule is about coordination, i.e., if two views encode the same fields in color and use more than 50% of the same fields, they will be coordinated by the *brushing* ($score = 6.25$). For example, as described in Figure 7 (2), Views (C) and (D) have a total of 5 fields, with

4 being identical, so 80% of the fields are shared. Moreover, both of them present the field *Profitable?* using colors, so that they brush each other. **E3** mentioned, "*This coordination is necessary, since it can help users identify the related points in another view and thus explore the data efficiently*". **E4** pointed out, "*This is really useful if there are too many colors or objects in one view, like a scatterplot*".

Overview first, then details-on-demand. The last two rules are about the relative position of views. The first one shows that *Text* view (not *Text* table) tends to be on the top right, top left, or top of the other view types ($score = 5.5$). For example, Figure 7 (A) was arranged on the top left or top right of the other three views, as described in Figure 7 (3). **E1** said, "*Text view providing the statistical values works as an overview and attracts the audiences' attention. Hence, the overview should be at the top of a dashboard*". While **E2** and **E4** argued, "*it is correct to put the statistical Text view at the top, but the word cloud should not always be put at the top*". The last rule is that if View A encodes more fields than View B, then View A prefers to be on the bottom left, bottom right, or bottom of View B ($score = 5.5$). **E3** mentioned, "*the view with more fields tends to be more detailed than the view with fewer. According to Schneiderman's visualization mantra 'overview first, then details on demand' [49], the view with more fields should be at the bottom or on the right*". **E4** agreed, "*the view with fewer fields may have more coarse information, thus having a higher hierarchy at the top*". **E2** considered that "*the overview should be a simple view with fewer fields*". For example, Figure 7 (4) described that View (A) had the least fields, thus being arranged on top of other views. While View (D) encoded more fields than View (B), i.e., 4 vs. 3, then View (D) was put at the bottom right of View (B).

Limitations and lessons. We also received some feedback on those processed rules with low scores, which can be summarized into two main reasons, i.e., different personal

experiences and shortcut learning of the algorithm. From personal experience, experts may have different opinions on a few rules. For example, the rule, *Scatterplot should be put on the right-most*, received a score of 3.5 ± 1.73, denoted as the 6th rule. **E1** mentioned, "*I have previously used scatterplots to show specific details, such as the distribution relationship of certain variables in two selected cities. I tend to put the detailed view on the right. Thus, I agree with this extracted rule*". In contrast, **E3** often used T-SNE to reduce high-dimensional data to a two-dimensional scatterplot to overview the system. Therefore, he thought "*the scatter plot should be on the left or top rather than right-most*". Another rule with a higher score variance of 1.91 is that *the view raising brushing should have a width or height of 3 or 4*, denoted as the 7th rule. **E4** has developed systems that use slender panels on the left or wide panels on the top. Thus, she thought it was reasonable for views raising *brushing* to have a larger width and height. While **E3** mentioned that "*the view proposing brushing should be a high-level view and therefore have a smaller size. 3 or 4 is too large for such a view*". Another problem is that the decision rule approach learns some highly-correlated relationships between attributes but lack semantics for humans to interpret. For example, for the rule (*If View A and View B encode the same field in color, then the height of View A or View B is 3*), **E2** commented that "*DMiner may derive this rule from the rules mentioned before (the 3rd rule and the 7th rule). However, it is challenging for humans to understand such a derivation or inference and make sense of this rule*".

5.2 User Study

We conducted a user study with 12 participants to compare 6 methods of designing MV dashboards and collect their feedback about the effectiveness of our approach for recommending appropriate MV dashboards.

5.2.1 Study Setup

Participants. We recruited 12 participants (9 females and 3 males, ages 22 to 27, with an average of 24.8) by posting online advertisements on social media platforms (e.g., email and microblogging websites). They are all well-educated (7 PhD students, 3 MSc students, 1 research assistant, and 1 data analyst) and from diverse backgrounds, including visualization (4), data science (3), finance (2), human-computer interaction (2) and recommendation system (1). None of them has color blindness. All the participants are experienced in the data analysis according to their self-reports on a 7-point Likert scale ($\mu = 5.25, \sigma = 0.45$), where 1 indicates "no experience" and 7 represents "highly experienced". Each participant was compensated with $10 after finishing the user study.

Dashboards and Baseline Methods. We selected 15 dashboards from the test dataset covering almost all the mark types (10/13) and various topics (e.g., business, COVID-19, and patient analysis). The number of views ranges from 3 to 8, obeying the distribution in Figure 3 (A). To evaluate the effectiveness, we built a comparative group designed by five experienced designers (3 females and 2 males, aged 27 ± 2.74). Two are postdocs at the university, focusing on the visual analytics of computational social science and narrative storytelling, respectively. The remaining three designers are data analysts in automobile, outlet malls, and retail companies. They are all experienced in data analysis (with an average of 4.1 years) and dashboard design (with an average of 3 years). Each designer was randomly assigned three dashboards. They were given the views of each dashboard and started their design after becoming well familiar with the view content. Specifically, the design space of the dashboard was fixed at the common display size of 1080p (i.e., 1920 x 1080) to enable comparisons. All designers finished the design in around 1 hour with $15 compensation. In this study, we leveraged 6 methods to recommend appropriate designs for 15 dashboards. The 6 methods are as follows:

- *Default*: the *Default* dashboard designs recommended by Tableau [5]. Note that Tableau does not support the recommendation of coordination. Thus, the corresponding dashboards do not have coordination;
- *DMiner-Basic*: the dashboard designs recommended by *DMiner-Basic* using design rules concerning only the single-view data and encoding features;
- *DMiner-Partial*: the dashboard designs recommended by *DMiner-Partial* considering part of design rules, i.e., all black relationships in Figure 4, and ignoring those rules on interrelations between arrangement and coordination;
- *DMiner-Full*: the dashboard designs recommended by *DMiner-Full* considering all extracted design rules;
- *Human*: the original dashboard designs we crawled online, which are designed by general *Human* users; and
- *Designer*: the dashboard designs created by the recruited experienced *Designers*.

The above 6 methods were used to recommend dashboard designs for a given dashboard. All the generated dashboard designs were of consistent size (i.e., 1920 x 1080 pixels), except for *Human*, whose corresponding dashboard design was in the original size. For the selected 15 dashboards, it took designers around eight minutes to design a dashboard. Our three methods speeded up this process by a factor of eight. In addition, our three methods generated designs within five seconds for dashboards with fewer than six views. For simplicity, the resulting six dashboard designs are called *a dashboard group*. Therefore, we presented 15 dashboard groups for participants to evaluate their design effectiveness.

Procedure. The user study lasted about 1.5 hours, and we gained participants' consent for video recording the whole user study. During the user study, we first briefly introduced our work. Then, due to the long time cost of evaluating dashboard designs, we randomly split the 15 dashboard groups into 3 clusters (each with 5 dashboard groups). Each participant was asked to score the arrangement and coordination of 1 cluster (i.e., 5 dashboard groups). For each dashboard group, the 6 dashboard designs were ordered randomly and anonymously assigned to participants. Thus, each dashboard group was evaluated by 4 participants. Before participants started to evaluate each dashboard group, we introduced every dashboard view that needed to be arranged. Once participants were familiar with these views, they freely explored the dashboards for around ten minutes. After the free exploration, they further scored the dashboard
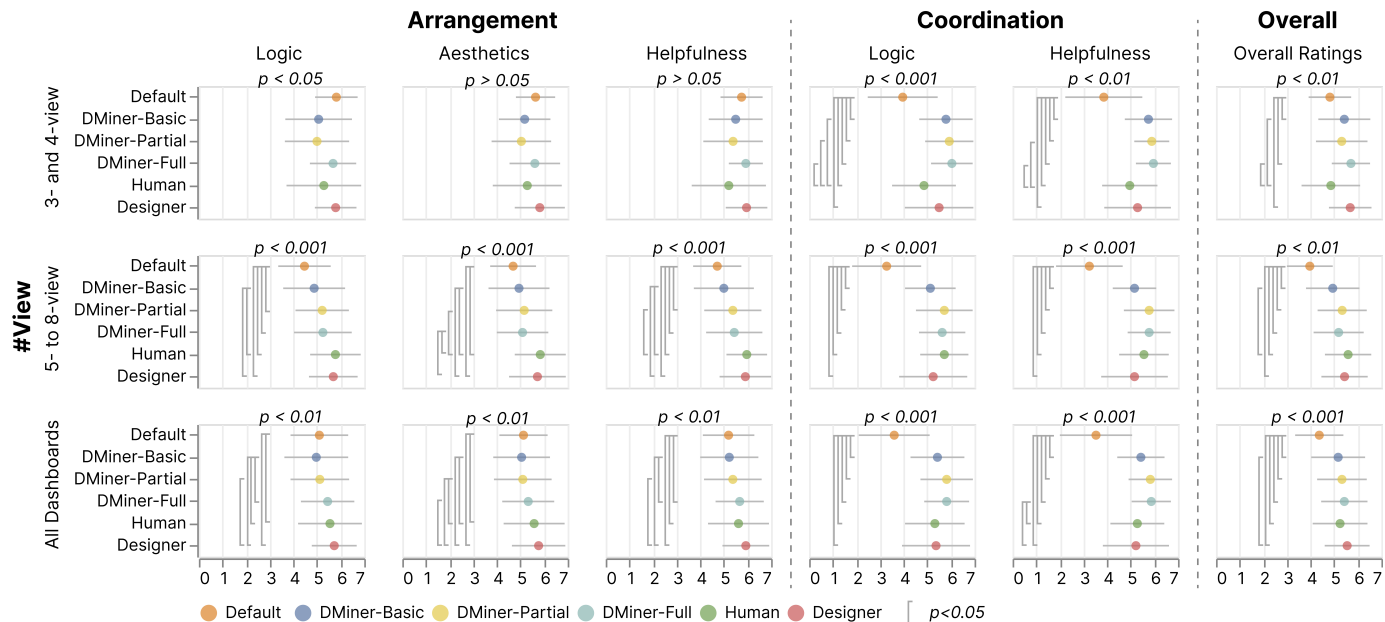
Figure 8. An overview of the study result. Specifically, each column describes the score of a metric in three clusters, i.e., the cluster considering all dashboards, the cluster with 3- and 4-view dashboards, and the cluster with 5- to 8-view dashboards. Six metrics cover 3 perspectives, i.e., arrangement, coordination, and overall (i.e., both). Each sub-plot presents the score of 6 methods regarding the corresponding metric and cluster. The error bars are 95% confidence intervals. ⌈ indicates the significant difference between a pair of methods ($p < 0.05$).

designs using a 7-point Likert scale. Participants were asked to evaluate their arrangement (including logic, aesthetics, and helpfulness), coordination (including logic and helpfulness), and overall, which refers to the previous studies [29], [50]. Specifically, the *logic* measures how well the arrangement and coordination follow human users' *logical* analysis workflow [51]. The *helpfulness* describes how well the arrangement and coordination help users analyze and explore data. The *aesthetics* evaluates how users perceive the visual appearances of the dashboard arrangement, while the *overall* score measures how well the dashboard has been designed concerning both the arrangement and coordination. All the participants were encouraged to report why they gave the corresponding scores in a think-aloud manner.

5.2.2 Result Analysis

This section introduces and discusses results from the user study, including the ratings and the corresponding qualitative feedback. We received positive feedback on the recommender of DMiner. Participants appreciated the automated dashboard designs in terms of both arrangement and coordination by our approach. Participant 11 (P11 for short) commented, "*I hope it can be integrated into my workflow, [...], I believe it will greatly facilitate my analysis process*". To analyze the results, we further split all dashboards into two *dashboard clusters* according to the number of views in dashboards, i.e., the dashboards with 3-4 views and the dashboards with 5-8 views. The rationale for the split stemmed from the feedback from our participants. They indicated that they cared more about the arrangement and coordination when the number of views is more than 4, since a well-organized dashboard can reduce their cognitive load of viewing many visualizations. We performed the one-way ANOVA to compare the six methods across six metrics and three dashboard clusters with different view

numbers (i.e., *3 and 4 views*, *5 to 8 views*, and *all*). LSD post-hoc tests were used if the scores obeyed the homogeneity of variance test; otherwise, Tamhane's T2 post-hoc tests were used [52]. Detailed information, including the average ratings, standard deviation, and post-hoc results, are presented in Figure 8. Figure 8 shows that all the other methods achieved better results than *Default* on all metrics, except for the 3- and 4-view dashboard arrangement. Specifically, *Default* mainly scored below 5, while the other five methods scored mostly above 5. In the following, we first introduce some interesting results regarding arrangement, coordination, and overall performance, then summarise these results, and finally discuss some recommendations that can be further improved.

Arrangement. For **logic**, Figure 8 shows our method *DMiner-Full* was not significantly different to *Designer* and *Human* regardless of the number of views. Specifically, it was slightly better than *Human* for 3- and 4-view dashboards. Moreover, our method *DMiner-Full* performed significantly better than *Default* for 5- to 8-view dashboards ($p < 0.01$). Figure 9 (A) provides one example, where (a) was created by *Human* and (b) was the recommended dashboard design by *DMiner-Full* (with an average score of logic as 5 and 5.25, respectively). Participants recognized that the arrangement logic of (b) was better than (a), since (b1) was more suitable than (a1) to act as a starting point to guide the data analysis. Furthermore, *DMiner-Full* achieved better arrangement logic than *Default* for dashboards with 5 to 8 views. Participants acknowledged that the arrangement logic was essential for dashboards with many views, while *Default* arranged the views alphabetically by view name, leading to worse logic that confused participants. For **aesthetics**, our method *DMiner-Full* offered a slightly better visual aesthetic design than *Default*. Participants appreciated that *DMiner-Full* set different sizes for different views,
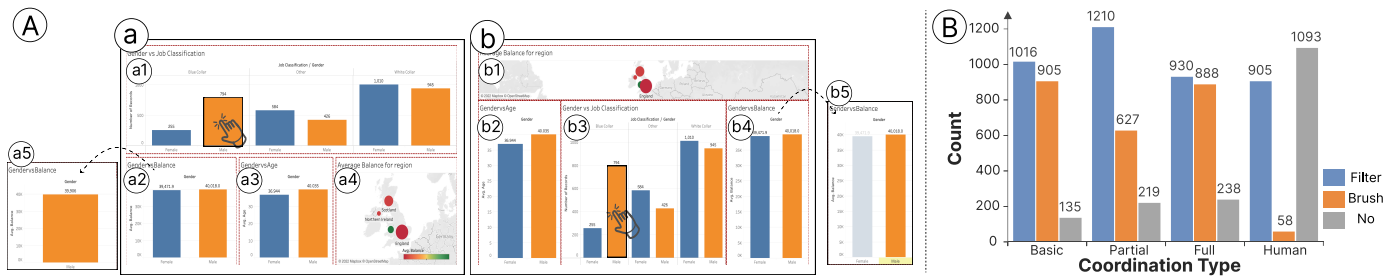
Figure 9. (A) is a case identified in our user study, and (B) is the coordination type distribution. In (A), (a) and (b) were the dashboards created by *Human* and our method *DMiner-Full*, respectively. Our recommended dashboard was highly rated due to outstanding arrangement logic and coordination type (see (a5) and (b5)). Participants recognized that (b1) was more suitable than (a1) to act as a starting point to explore the dashboard. For the same view pair, our recommended coordination type was *brushing*, which kept the context for comparison. However, *Human* chose *filtering*, which led to a change in the display and confused the users. In (B), our three recommenders (i.e., *DMiner-Basic*, *DMiner-Partial* and *DMiner-Full*) suggested more coordination than *Human*.

which was better than *Default* which set all views with equal width and height. P8 mentioned that "*when there are too many views, there should be a dominant one telling me what it is attempting to express. The exactly-equal size confuses me, and it is challenging for me to identify what I should pay attention to first*". Moreover, participants suggested that arrangement aesthetics was less important compared to arrangement logic for facilitating data analysis. Thus, for **helpfulness**, Figure 8 shows observations similar to the logic.

Coordination. Figure 8 shows that all our methods (i.e., *DMiner-Full*, *DMiner-Partial*, and *DMiner-Basic*) offered significantly better coordination than *Default* in terms of logic and helpfulness, regardless of the number of views (with all $p < 0.001$). Furthermore, *DMiner-Partial* and *DMiner-Full* performed similarly to *Human* and *Designer* in 5- to 8-view dashboards, and were significantly better than *Human* in 3- and 4-view dashboards (with all $p < 0.05$). Participants all confirmed the indispensability of the co-ordination among views in a dashboard. P1 mentioned, "*Without coordination, it is challenging for me to identify the related data across views and focus the data of interest*". As shown in Figure 9 (B), compared to *Human*, participants appreciated that our methods were equipped with more brushing or filtering between views for dashboards, as it provided a stronger power of exploration. Moreover, P5 and P11 appreciated the coordination type recommended by our approach in some cases. They mentioned, "*Filtering can help identify relevant information efficiently in those views with many visual elements, such as text tables with thousands of lines of text. Brushing works better for those views without visual clutter, since it maintains context and enables comparisons*". For example, as shown in Figure 9 (A), (a5) was the filtered result of (a2), while (b5) was the brushed result of (b4). It is clear to see that (b5) well preserved the context before and after the interaction. That is why our three methods gained the highest coordination score for 3- and 4-view dashboards.

Overall. From Figure 8, our method *DMiner-Full* achieved significantly better results than *Default* regardless of the number of views (with all $p < 0.001$). Moreover, compared to *Human* and *Designer*, the average scores obtained by *DMiner-Full* were similar without significant differences, except that *DMiner-Full* performed significantly better than *Human* for 3- and 4-view dashboards ($p < 0.01$). Participants mentioned that the arrangement of views in dashboards

was slightly more important than the coordination for data analysis, especially for dashboards with more views. For 3- and 4-view dashboards, *DMiner-Full* was significantly better than *Human*, as *DMiner-Full* can achieve similar arrangements as *Human*, but with much better and more coordination. Regardless of the number of dashboards, our method *DMiner-Full* achieved similar performance to *Designer*.

Summary. In summary, *DMiner-Full* performed significantly better than *Default* for dashboards with more views, and similarly to *Human* and *Designer* without significant differences. It was the best among our three methods, followed by *DMiner-Partial*, and then *DMiner-Basic*. With an increased number of views, *DMiner-Full* and *DMiner-Partial* always performed better than *DMiner-Basic*, demonstrating the importance of the pairwise-view relationship in modeling complex dashboards. Moreover, no significant difference between *Human* and *Designer* in 6 metrics was observed, indicating the quality of our collected dashboard dataset. Figure 10 shows an example where *Human* and *Designer* had different designs given the same views, where (A) was created by *Designer* and (B) was created by *Human*. *Designer* regarded (A1) as an overview to analyze the bike trips every day in a week, while *Human* adopted (B1) as the overview to analyze the most common trips that bike riders like. Participants recognized that they had different analysis focus, and both were designed with plausible arrangement and coordination (both with average scores larger than 5.75).

Limitations. DMiner occasionally provides some recommendations that can be further improved. According to participants' feedback, the underlying reasons for it stems from two perspectives, i.e., without considering the number of visual elements within each view and the semantic meaning of data. Take Figure 10 as an example, where the MV dashboard designed by *Designer* (A), *Human* (B) and our method *DMiner-Full* (C) are shown. Compared with Figure 10 (A2) and (B1), the same view in the dashboard design recommended by our approach (Figure 10 (C2)) does not have enough space to show the heatmap. The underlying reason for such a result is that DMiner does not explicitly consider the number of visual elements (i.e., many rectangles in the heatmap) due to the lack of underlying data tables. Also, a few participants pointed out that Figure 10 (C) placed the line chart showing the "Peak in Winter" above the line chart displaying the "Peak in Summer", which was
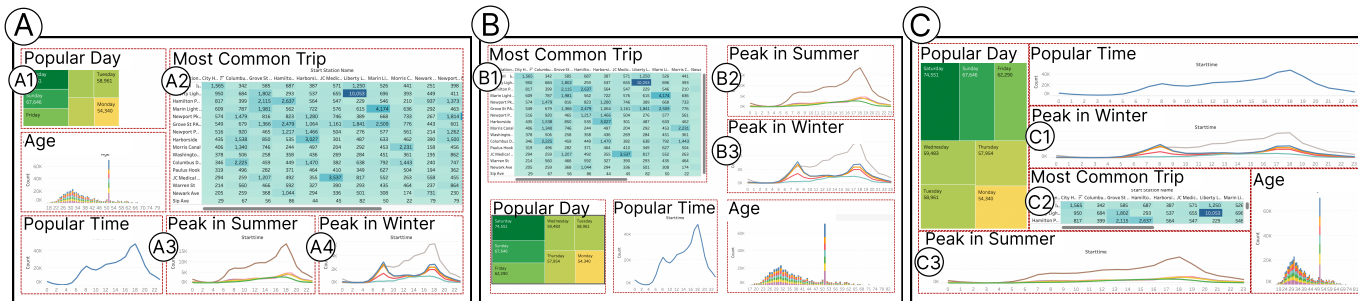
Figure 10. A case identified in our user study. (A) is created by *Designer*, (B) is created by *Human*, while (C) is created by our method *DMiner-Full*. (A) and (B) received similar scores in all metrics (with a difference of less than 0.5), though they have different designs with different analysis focus. *DMiner-Full* in (C) received a lower score, and our participants pointed out that the arrangements should be improved from two perspectives, i.e., the size of (C2) should be equipped with more space like (A2) and (B1), and (C1) about "winter" should be above (C3) about "summer".

contradictory to their expectation, i.e., the line chart for "Summer" should be positioned above that for "Winter". Unfortunately, such semantic meanings of the data are not considered in DMiner.

6 DISCUSSION

In this section, we further discuss a few issues that need further clarification regarding DMiner.

High-quality multiple-view dashboard dataset construction. Recent years have witnessed an increasing interest in applying AI and machine learning (ML) for visualization research [46], [53], where high-quality datasets are fundamental to making the AI/ML-based approaches work well in practice. However, little research has been dedicated to constructing large-scale, high-quality multiple-view dashboard datasets. In this paper, we construct a dashboard dataset by crawling GitHub repositories. However, the designers of the dashboards published on GitHub may have different levels of visualization expertise. Our results in Section 5.2.2 have shown that our collected dashboard designs (i.e., *Human* in Figure 8) are not significantly different to those designed by *Designer* in Figure 8. We believe that the quality of the dashboard designs can be further improved. For example, appropriate dashboard designs also depend on other important factors like target users, user intention, and analysis tasks, which, however, are not included in our current dashboard dataset. Thus, we would encourage the whole visualization community to contribute more high-quality and informative dashboard datasets.

Human agency vs. machine automation. DMiner proposes a data-driven framework to mine design rules and automatically recommend dashboard designs. Our evaluation in Section 5.2.2 has shown that the dashboard designs recommended by DMiner (i.e., DMiner-Full) are not significantly different from those created by experienced designers. Recent research on visualization recommendations has tried to balance human agency and machine automation by keeping humans in the loop [54]. For dashboard design recommendations, it is worth further exploring how the human agency can be involved in dashboard design mining. For instance, it will be interesting to investigate how expert knowledge can be incorporated into refining the extracted design rules for dashboards and further improve the quality of recommended dashboards.

DMiner for dashboards created by other software or packages. In this paper, DMiner extracts dashboard design rules from our collected dashboard datasets created by Tableau, one of the most popular software for creating multiple-view dashboards, and is further evaluated on only Tableau dashboards. However, DMiner can be extended to dashboards created using other software or packages. For example, the source files of dashboards created by Power BI [6] also specify visual coding, data operation, data types, and arrangement and coordination between views in a dashboard, which is quite similar to those of Tableau. With a proper data parsing module, DMiner can also work for dashboards by Power BI. However, this requires considerable research and engineering efforts to improve the interoperability, since different visualization software and packages have different methods for specifying and rendering visualizations [55]. With more multiple-view dashboard datasets available in the future, we would like to extend DMiner further to various dashboard datasets.

More factors for dashboard design. DMiner has considered a series of features, including data and visual encoding characteristics, for designing dashboards in terms of the arrangement and coordination of its views. However, our evaluation result in Section 5 has demonstrated its effectiveness. However, there exist other factors that warrant future research. As pointed out in Section 5.2.2, the semantic meaning of the visualized data can also influence the arrangement and coordination of dashboard views. For example, it is better to arrange the views showing the data of different seasons in the order of spring to winter (Section 5.2.2). Also, the decision rules of DMiner focus on the effects of single features, and it will be interesting to further explore the combined effects of multiple feature types in a single decision rule for dashboard design.

7 CONCLUSION

This paper proposes DMiner, a data-driven framework for mining dashboard design and recommending appropriate arrangement and coordination for multiple-view dashboards. Building upon our Tableau dashboard dataset collected from GitHub, DMiner extracts a series of relevant features, i.e., single-view and pair-view features in data and encoding, arrangement and coordination. With these features, DMiner employs a decision rule approach to mine the

design rules from the collected Tableau dashboard dataset. Further, a recommender is proposed to recommend appropriate arrangements and coordination of dashboard views. We conduct an expert study and a user study to evaluate the effectiveness of DMiner. The expert study demonstrates that the extracted design rules are reasonable and align well with the design practices of experts. The user study further confirms that the dashboard designs recommended by DMiner, considering all design rules, are not significantly different from those of experienced designers. They are also significantly better than those using the default settings of *Tableau* in both arrangement and coordination. In summary, as the first work for automating dashboard arrangement and coordination, DMiner is appreciated by visualization experts and study participants who need to design and use dashboards for data analysis. In the future, we hope to further improve the efficiency of our recommender for dashboards with more views. It is also promising to regard DMiner as a feature and integrate it into existing visualization authoring tools, such as Voyager [22], to help designers in their daily work.

ACKNOWLEDGMENTS

REFERENCES

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

[2] C. Forsell and J. Johansson, "An heuristic set for evaluation in information visualization," in *Proceedings of the 2010 International Conference on Advanced Visual Interfaces*, 2010, p. 199–206.

[3] R. Sadana and J. Stasko, "Designing multiple coordinated visualizations for tablets," in *Proceedings of the Computer Graphics Forum*, vol. 35, no. 3, 2016, pp. 261–270.

[4] R. Chen, X. Shu, J. Chen, D. Weng, J. Tang, S. Fu, and Y. Wu, "Nebula: a coordinating grammar of graphics," *IEEE Transactions on Visualization and Computer Graphics*, 2021 (Early access).

[5] "Tableau," https://help.tableau.com/current/pro/desktop/en-us/gettingstarted_overview.htm, [Online; accessed 2022-08-23].

[6] "Data Visualization | Microsoft Power BI," https://powerbi.microsoft.com/en-us/, [Online; accessed 2022-08-23].

[7] A. Wu, Y. Wang, M. Zhou, X. He, H. Zhang, H. Qu, and D. Zhang, "MultiVision: Designing analytical dashboards with deep learning based recommendation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 162–172, 2021.

[8] A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher, "What do we talk about when we talk about dashboards?" *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 682–692, 2018.

[9] B. Saket, D. Moritz, H. Lin, V. Dibia, C. Demiralp, and J. Heer, "Beyond heuristics: Learning visualization design," *arXiv preprint arXiv:1807.06641*, 2018.

[10] "Github," https://github.com, [Online; accessed 2022-08-24].

[11] J. C. Roberts, "State of the art: Coordinated & multiple views in exploratory visualization," in *Proceedings of the 5th International Conference on Coordinated and Multiple Views in Exploratory Visualization*, 2007, pp. 61–71.

[12] B. Bach, E. Freeman, A. Abdul-Rahman, C. Turkay, S. Khan, Y. Fan, and M. Chen, "Dashboard design patterns," *arXiv preprint arXiv:2205.00757*, 2022.

[13] Z. Qu and J. Hullman, "Keeping multiple views consistent: Constraints, validations, and exceptions in visualization authoring," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 468–477, 2017.

[14] R. Langner, U. Kister, and R. Dachselt, "Multiple coordinated views at large displays for multiple users: Empirical findings on user behavior, movements, and distances," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 608–618, 2018.

[15] M. Sun, A. Namburi, D. Koop, J. Zhao, T. Li, and H. Chung, "Towards systematic design considerations for visualizing crossview data relationships," *IEEE Transactions on Visualization and Computer Graphics*, 2021 (Early access).

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

[17] M. Sun, A. R. Shaikh, H. Alhoori, and J. Zhao, "SightBi: Exploring cross-view data relationships with biclusters," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 54–64, 2021.

[18] V. Dibia and Ç. Demiralp, "Data2Vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks," *IEEE Computer Graphics and Applications*, vol. 39, no. 5, pp. 33–46, 2019.

[19] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo, "VizML: A machine learning approach to visualization recommendation," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[20] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer, "Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 438–448, 2018.

[21] H. Li, Y. Wang, S. Zhang, Y. Song, and H. Qu, "KG4Vis: A knowledge graph-based approach for visualization recommendation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 195–205, 2021.

[22] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, "Voyager: Exploratory analysis via faceted browsing of visualization recommendations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 649–658, 2015.

[23] A. Key, B. Howe, D. Perry, and C. Aragon, "VizDeck: selforganizing dashboards for visual analytics," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 2012, pp. 681–684.

[24] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko, "Augmenting visualizations with interactive data facts to facilitate interpretation and communication," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 672–681, 2018.

[25] A. Tundo, C. Castelnovo, M. Mobilio, O. Riganelli, and L. Mariani, "Declarative dashboard generation," in *Proceedings of 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2020, pp. 215–218.

[26] D. Deng, A. Wu, H. Qu, and Y. Wu, "DashBot: Insight-driven dashboard generation based on deep reinforcement learning," *arXiv preprint arXiv:2208.01232*, 2022.

[27] J. Hullman, S. Drucker, N. H. Riche, B. Lee, D. Fisher, and E. Adar, "A deeper understanding of sequence in narrative visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2406–2415, 2013.

[28] Y. Kim, K. Wongsuphasawat, J. Hullman, and J. Heer, "GraphScape: A model for automated reasoning about visualization similarity and sequencing," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 2628–2638.

[29] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang, "DataShot: Automatic generation of fact sheets from tabular data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 895–905, 2019.

[30] D. Shi, X. Xu, F. Sun, Y. Shi, and N. Cao, "Calliope: Automatic visual data story generation from a spreadsheet," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 453–463, 2020.

[31] H. M. Al-maneea and J. C. Roberts, "Towards quantifying multiple view layouts in visualisation as seen from research publications,"

in *Proceedings of the 2019 IEEE Visualization Conference (VIS)*, 2019, pp. 121–121.

[32] X. Chen, W. Zeng, Y. Lin, H. M. Ai-Maneea, J. Roberts, and R. Chang, "Composition and configuration patterns in multiple-view visualizations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 1514–1524, 2020.

[33] L. Shao, Z. Chu, X. Chen, Y. Lin, and W. Zeng, "Modeling layout design for multiple-view visualization via bayesian inference," *Journal of Visualization*, vol. 24, no. 6, pp. 1237–1252, 2021.

[34] M. Lu, C. Wang, J. Lanir, N. Zhao, H. Pfister, D. Cohen-Or, and H. Huang, "Exploring Visual Information Flows in Infographics," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–12.

[35] L. Bartram and C. Ware, "Filtering and brushing with motion," *Information Visualization*, vol. 1, no. 1, pp. 66–79, 2002.

[36] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A grammar of interactive graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 341–350, 2016.

[37] E. R. Tufte, "The visual display of quantitative information," *The Journal for Healthcare Quality*, vol. 7, no. 3, p. 15, 1985.

[38] J. Roberts, "On encouraging multiple views for visualization," in *Proceedings of the 1998 IEEE Conference on Information Visualization.*, 1998, pp. 8–14.

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

[40] A. Wu, W. Tong, H. Li, D. Moritz, Y. Wang, and H. Qu, "ComputableViz: Mathematical operators as a formalism for visualisation processing and analysis," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–15.

[41] S. Greco, B. Matarazzo, and R. Słowiński, "Decision rule approach," in *Multiple criteria decision analysis*. Springer, 2016, pp. 497–552.

[42] C. Singh, K. Nasseri, Y. S. Tan, T. Tang, and B. Yu, "imodels: a python package for fitting interpretable models," p. 3192, 2021.

[43] J. H. Friedman and B. E. Popescu, "Predictive learning via rule ensembles," *The Annals of Applied Statistics*, vol. 2, no. 3, pp. 916–954, 2008.

[44] "Binary vector - Wikipedia," https://en.wikipedia.org/wiki/Binary_vector, [Online; accessed 2022-08-23].

[45] Y. Ming, H. Qu, and E. Bertini, "RuleMatrix: Visualizing and understanding classifiers with rules," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 342–352, 2018.

[46] A. Wu, Y. Wang, X. Shu, D. Moritz, W. Cui, H. Zhang, D. Zhang, and H. Qu, "AI4VIS: Survey on artificial intelligence approaches for data visualization," *IEEE Transactions on Visualization and Computer Graphics*, 2021 (Early access).

[47] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis, "SEEDB: Efficient data-driven visualization recommendations to support visual analytics," in *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, vol. 8, no. 13. NIH Public Access, 2015, p. 2182.

[48] R. Ding, S. Han, Y. Xu, H. Zhang, and D. Zhang, "QuickInsights: Quick and automatic discovery of insights from multi-dimensional data," in *Proceedings of the 2019 ACM SIGMOD International Conference on Management of Data*, 2019, pp. 317–332.

[49] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *The Craft of Information Visualization*, 2003, pp. 364–371.

[50] V. S. Smith, "Data dashboard as evaluation and research communication tool," *New Directions for Evaluation*, vol. 2013, no. 140, pp. 21–45, 2013.
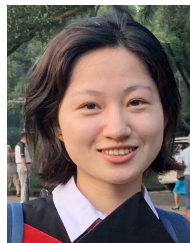
[51] D. Dowding and J. A. Merrill, "The development of heuristics for evaluation of dashboard visualizations," *Applied Clinical Informatics*, vol. 9, no. 03, pp. 511–518, 2018.

[52] S. Lee and D. K. Lee, "What is the proper way to apply the multiple comparison test?" *Korean journal of anesthesiology*, vol. 71, no. 5, pp. 353–360, 2018.

[53] Q. Wang, Z. Chen, Y. Wang, and H. Qu, "A Survey on ML4VIS: Applying machinelearning advances to data visualization," *IEEE Transactions on Visualization and Computer Graphics*, 2021.

[54] Y.-R. Cao, X.-H. Li, J.-Y. Pan, and W.-C. Lin, "VisGuide: User-oriented recommendations for data event extraction," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–13.

[55] A. Satyanarayan, B. Lee, D. Ren, J. Heer, J. Stasko, J. Thompson, M. Brehmer, and Z. Liu, "Critical reflections on visualization authoring systems," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 461–471, 2019.

Yanna Lin is currently a PhD candidate in Computer Science and Engineering at the Hong Kong University of Science and Technology (HKUST). Her main research interests are data visualization and human-computer interaction. She received her BEng from Sun Yat-sen University. For more details, please refer to https://yannahhh.github.io/.

Haotian Li is currently a PhD candidate in Computer Science and Engineering at the Hong Kong University of Science and Technology (HKUST). His main research interests are data visualization, visual analytics, human-computer interaction and online education. He received his BEng in Computer Engineering from HKUST. For more details, please refer to https://haotian-li.com/.

Aoyu Wu is a Ph.D. student in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology (HKUST). He received his B.E. and M.E. degrees from HKUST. His research interests include data visualization and human-computer interaction. For more details, please refer to https://wowjyu.github.io/.

Yong Wang is currently an assistant professor in School of Computing and Information Systems at Singapore Management University. His research interests include data visualization, visual analytics and explainable machine learning. He obtained his Ph.D. in Computer Science from Hong Kong University of Science and Technology in 2018. He received his B.E. and M.E. from Harbin Institute of Technology and Huazhong University of Science and Technology, respectively. For more details, please refer to http://yong-wang.org.

Huamin Qu is a chair professor in the Department of Computer Science and Engineering (CSE) at the Hong Kong University of Science and Technology (HKUST) and also the director of the interdisciplinary program office (IPO) of HKUST. He obtained a BS in Mathematics from Xi'an Jiaotong University, China, an MS and a PhD in Computer Science from the Stony Brook University. His main research interests are in visualization and human-computer interaction, with focuses on urban informatics, social network analysis, E-learning, text visualization, and explainable artificial intelligence (XAI). For more information, please visit http://huamin.org/.