Waltzboard: Multi-Criteria Automated Dashboard Design for Exploratory Analysis

Category: Research

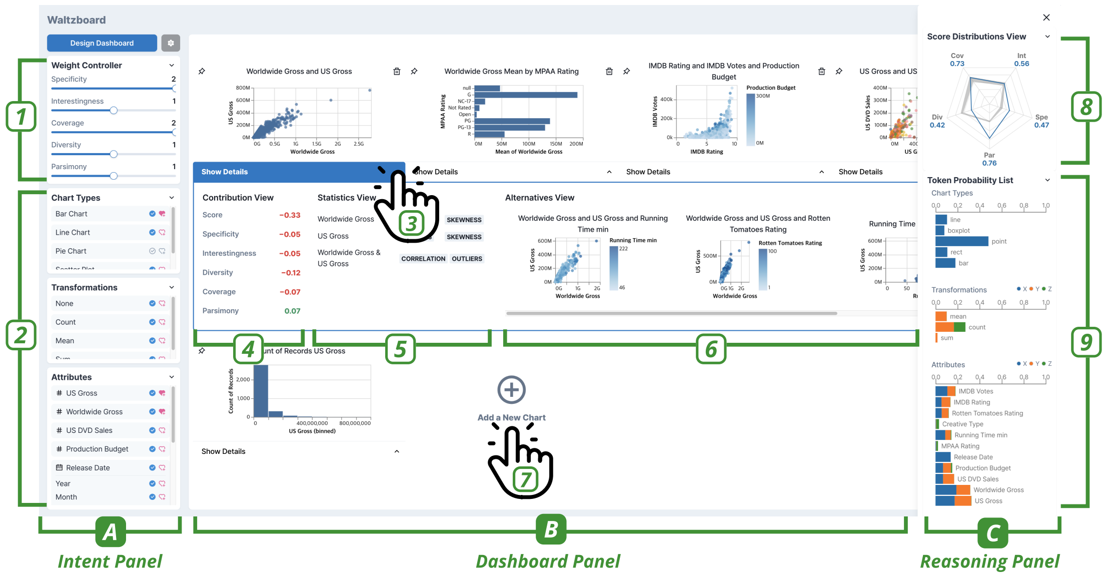

Figure 1: The user interface of *Waltzboard*. A The Intent Panel allows the user to flexibly express their intent by tuning the weights of five measures (1) and prioritizing chart types of attributes of interest (2). B The Dashboard Panel shows a dashboard generated automatically with the contribution of each chart to the measures (4 and 5) and its alternatives (6). C The Reasoning Panel provides the details of the automatic design process, e.g., the score of the current design compared to alternatives.

Abstract

We present *Waltzboard*, an automated dashboard design system for exploratory data analysis. Despite the benefit of dashboards, which provides a glanceable overview of data, previous dashboard design systems often require precomputation, such as training deep-learning models, and do not adapt effectively to changes in the user's intent during data analysis, hindering quick and flexible data exploration. To overcome these challenges, we introduce a dashboard evaluation framework that quantifies how a dashboard describes data in terms of five key measures: Specificity, Interestingness, Diversity, Coverage, and Parsimony. We then present a three-phase search algorithm designed to efficiently explore dashboard designs without the need for precomputation. Finally, we present a user interface that allows the user to dynamically build their own intent and reason for the design process. The result of our performance benchmark and user study demonstrates that *Waltzboard* not only designs a more effective dashboard within seconds but also supports flexible exploratory data analysis to meet diverse analytic needs.

Keywords: Dashboard Design, Exploratory Data Analysis, Automated Data Analysis

1 Introduction

Dashboards allow people to quickly glance at data [21], and this nature has led to their widespread use across diverse domains [3, 18, 36, 47, 54]. However, designing a dashboard entails complex cognitive processes such as understanding data characteristics, determining the attributes to include, and selecting suitable visual encodings. These processes often require significant time and labor, preventing analysts themselves from benefiting from the glanceable overview that dashboards are meant to provide.

Recently proposed deep learning-based systems, e.g., Multivision [51] and Dashbot [15], aim to streamline manual dashboard design, but they have prerequisites such as the availability of domain-specific datasets [51] or lengthy training time [15], undermining the benefit of dashboards. Additionally, their end-to-end architecture prevents interactive user intervention and customization based on intent. While rule-based systems like MEDLEY [35] are less strict to such prerequisites by employing predefined dashboard templates and allowing the user to engage in the dashboard design process, their reliance on fixed numbers of templates limits flexibility, particularly when the user intent is ambiguous in exploratory analysis.

We introduce *Waltzboard*, an automated dashboard design system tailored for exploratory data analysis (EDA). *Waltzboard* is *flexible*; it allows the user to build their own intent interactively and reflects the intent in the automated design process. *Waltzboard* is also *fast*; it does not require long precomputation nor processing time, making it well-suited for a cold-start analysis of multidimensional datasets. Finally, *Waltzboard* is *interpretable*; it helps the user understand why a certain dashboard is chosen and what alternatives are available.

In the rest of this paper, we present a framework for evaluating the effectiveness of a dashboard based on our survey on previous EDA systems and dashboard design guidelines [5,7,14,15,23,27,35–37,46,49–51]. Then, we elaborate on the design of *Waltzboard* built upon our evaluation framework and a simple yet efficient dashboard search algorithm. Finally, we evaluate our system by conducting a performance benchmark and a user study. We also make *Waltzboard* open-source [1] to facilitate future research on automated dashboard design. In summary, the contributions of our work are as follows:

[1]A URL will be included upon publication

- We define a framework that evaluates a dashboard in terms of five effectiveness measures derived from previous studies,
- We design and implement an automated dashboard design system *Waltzboard*, built upon the evaluation framework and an efficient dashboard search algorithm, and
- We report on a performance benchmark and a qualitative user study to investigate if and how *Waltzboard* facilitates the dashboard design process and exploratory data analysis.

2 Related Work

2.1 Visualization Recommendation Systems

We could categorize previous visualization recommendation systems into two groups: 1) systems recommending *how* to present the data by suggesting an effective encoding (e.g., visual marks and channels) under the given constraints, and 2) systems recommending *what* to present by suggesting a subset of data worth being visually analyzed (e.g., attributes or data facts). Note that the two themes are intertwined; the choice of how is often affected by the what, and the contributions of previous work often span both themes.

Systems focusing on *how* help the user select appropriate visual marks and channels. Examples include Voyager [49,50], which uses heuristics to recommend effective charts based on user constraints, and Draco [33], which formalizes design knowledge as constraints. Recent approaches like VizML [24] use deep learning to model visualization design from user-created examples.

Systems recommending *what* help the user identify interesting data insights by ranking visualizations generated from attribute pairs. SeeDB [43] uses deviation-based metrics, while Foresight [14] employs statistical features for ranking.

However, these systems are limited to single visualizations and cannot effectively recommend multi-visualization dashboards, as they neglect relationships between visualizations. We address this limitation by proposing a dashboard search algorithm that treats design as a multivariable optimization problem, considering the complementary nature of different visualizations.

2.2 Automated Dashboard Design Systems

Closely related to this work, there exist systems that recommend multiple visualizations as a dashboard to convey important facets of datasets. These dashboard design systems can be further categorized into two types according to the type of dashboards they generate according to Bach et al. [5]: 1) systems for curated dashboards and 2) systems for data collection dashboards.

Curated dashboard design systems support author-driven storytelling [39] with specific intent [5], e.g., a message to deliver with a deeper understanding of data. Examples include LADV [32], which converts hand-drawn sketches to dashboards, and DMiner [30], which recommends visualization arrangements. On the other hand, our work designs data collection dashboards to support reader-driven storytelling [39] and visualize large volumes of data [5], targeting the user who wants to explore the data using dashboards is not expected to have a concrete understanding of the data. There are a few systems that support EDA using dashboards, and we categorize them into two types: rule-based and deep learning-based.

Examples of rule-based dashboard design systems include DataShot [45] and VizDeck [25], where the user can generate dashboards by choosing top-ranked charts sorted by predefined rules (e.g., statistics). Recent systems like MEDLEY [35] and BOLT [40] incorporate user intent through dashboard templates derived from real-world usage patterns. In contrast, deep learning-based systems take a data-driven approach to dashboard design. Examples include Multivision [51], which uses LSTM neural networks for automated design, and Dashbot [15], which employs reinforcement learning to simulate human exploration patterns. However, these systems require training data [51] or training phases [15], limiting their availability and user interaction. While rule-based systems offer better control and speed, their predefined templates constrain adaptation to changing analytic intent [3].

In our work, we present a more general and flexible approach, allowing the user to express their intent more flexibly by describing their preference on the dashboard as a combination of five aspects (e.g., focusing on diversity and interestingness simultaneously). Furthermore, our system does not resort to a small set of dashboard templates or large-scale training datasets, leading itself to EDA.

2.3 User Intent in Visualization Systems

To be effective, many automated visualization systems consider the user's behaviors, goals, and tasks, which are often called "user intent," while there is no unified definition of the user intent, and the level of the user intent may vary depending on the context. Although there have been several systems that captured the user intent in an implicit way, for example, by inferring the intent from interaction logs [19, 22, 41], most of the previous automated visualization systems have the user explicitly express their intent [42].

One of the straightforward ways to capture the user intent is to have the user enter a partial visualization specification by choosing data attributes or encodings. For instance, Voyager 2 [50] uses wildcards (i.e., empty attributes that the system has to interpolate) for attribute specification. However, this approach only allows the user to describe their intent using low-level visualization specifications, limiting their ability to express their high-level goals or tasks. Recent systems have attempted to directly incorporate higher-level intent; for example, MEDLEY [35] derives four dashboard analytic intents, of which the user can choose one and Intentable [11] identifies four types of caption intent from large-scale visualization caption corpus.

However, defining or inferring user intent becomes particularly challenging in exploratory data analysis. This is because analysts often either lack clear intent or their intent continuously evolves across a broad spectrum [3,7]. In our work, to capture the user intent and provide more effective recommendations, we choose an explicit and bottom-up approach; we identify five measures that prioritize different aspects of dashboard designs from prior work and allow the user to express their intent by adjusting the weights between them.

3 Overview

Instead of providing the user with predefined intent options, our system allows the user to build their own intents in a bottom-up manner. In EDA, intents are notably diverse and continuously evolve as the analysis continues [7]. Considering this, it is hard to evaluate how the current dashboard is effective to the user's intent. Also, it means that it is unable to compare dashboards in search space.

To address this challenge, instead of proposing a single definition of user intent, we present five fundamental building blocks of the intent: *Specificity*, *Interestingness*, *Diversity*, *Coverage*, and *Parsimony*. These components form an evaluation framework for assessing dashboard effectiveness in exploratory analysis. On the *Waltzboard* interface, the user can dynamically adjust weights ($w\cdot$) between these measures to construct their personalized intent. In the following section, we demonstrate how the user can express their intents through these weights and measures and present design goals that support this analytical workflow.

3.1 Usage Scenario

Suppose Amy, a data analyst working in a film investment company, explores a movie dataset to understand the film industry.

Because this is her first time seeing the dataset, she wants to see a dashboard that provides an overview. To express such intent, she uses an intent panel (Fig. 1 A). She first increases the weights for coverage and diversity scores using the slider controls of the Weight Controller (Fig. 2 1). Subsequently, she turns off the checkboxes

(✓ →○) for `Pie Chart` since she is less interested in comparing a part-to-whole relationship (Fig. 2(2)). She also excludes certain data transformations (✓ →○) such as `Max` and `Min` as she wants to see the overall trend rather than extremes.

After expressing her needs, she clicks on the **Design Dashboard** button, and *Waltzboard* generates a dashboard with five charts in seconds in a dashboard panel (Fig. 1 B). Looking at the dashboard, she finds out there is only one chart that shows the profits of movies, which is actually of interest to a film investor. She now wants a dashboard that shows related to profits.

To express her refined intent, she increases the weight of specificity to 2 and decreases that of coverage back to 1 again. To articulate her interest in attributes related to profit, she clicks on heart icons (♡ →♥) next to attributes `US Gross`, `Worldwide Gross`, and `Production Budget` (Fig. 2(2)). She then clicks on the **Design Dashboard** button to refresh the dashboard with her new intent.

Since she lowered the weight for coverage, she worries that the new dashboard might not include enough attributes. To check this, she opens up a reasoning panel (Fig. 1 C) and realizes the current dashboard outperforms the average in terms of both specificity and coverage (Fig. 2(3)). She also notices that the search process gave a high sampling probability to the attributes that she prefers (Fig. 2(4)), which further strengthens her confidence in the design process.

Looking into the dashboard, she finds a heatmap named "Count of US Gross and Worldwide Gross" and becomes curious about why such a chart was included. She opens up the Detail View for the histogram by clicking on a "Show Details" button below the histogram (Fig. 1(3)) and finds out that the chart was considered interesting as it shows a correlation between the two attributes (Fig. 2(5) and (6)). She changes the histogram to a scatterplot to inspect individual movies with high gross by choosing the scatterplot on the Alternatives View (Fig. 2(7)). Then she notices charts for `Production Budget` are insufficient and wants to add more. She opens up the New Chart View (Fig. 2(8) and (9)) and chooses one from the recommendations, which shows the relationship between `Running Time` and `Worldwide Gross`, continuing her analysis.

3.2 Design Goals

We carefully surveyed the lessons learned from the previous dashboard design systems [15, 35, 51] as well as the guidelines on exploratory data analysis [3, 7, 31, 46]. As a result, we set three design goals that we considered during the design of *Waltzboard*.

DG1: Interactively reflect the user's intent. Previous studies pointed out that exploratory analysis often has no obvious analytic goals [3, 7, 31], and analysts refine their analytic intent as the analysis progresses [50]. However, previous fully automated dashboard design systems are unable to ensure the user's varying analytic intent during the design process [15, 51]. Inspired by these limitations, we aim to make our system able to pace with the user's changing analytic needs. For example, in the early phase of analysis, the user may want to obtain an overview of the dataset without prior knowledge. In this case, they could give a higher weight to diversity and coverage scores. On the other hand, as their analytic needs shift from breadth-oriented search to depth-oriented, they could increase the weight for specificity and add certain attributes or chart types to the user preference set P to see a dashboard focusing on a particular aspect of the dataset.

DG2: Enable cold-start analysis without long precomputation. One of the notable advantages of dashboard-based exploratory visual analysis is its ability to provide a quick, glanceable overview [5, 36], even for an unseen dataset, i.e., cold-start analysis. Nevertheless, previous automated dashboard design systems do not effectively support such analysis, often requiring the user to perform various forms of precomputation, such as necessitating a training dataset [51], relying

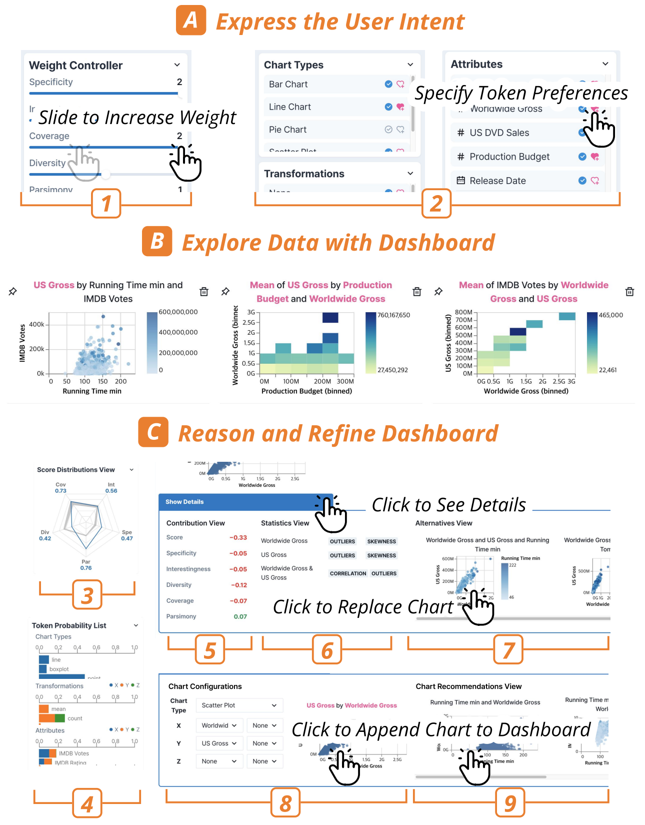

Figure 2: Usage scenario of *Waltzboard*. A The user begins by expressing their intent using the Weight Controller (1) and Token Controllers (2). B *Waltzboard* instantly designs a dashboard based on the user's intent. C The Score Distribution View (3) allows the user to assess the effectiveness of the current dashboard, while the Token Probability List (4) displays the sampling probabilities of chart tokens. The user can see why a certain chart is chosen through the Contribution View (5) and the Insight View (6). If needed, the user can replace a chart using the Alternatives View (7). The user can append a chart manually using the Chart Configuration Panel (8) or automatically via the Chart Recommendations View (9).

on training deep learning models [15], or crafting visualization templates [35]. With this goal in mind, we present a dashboard search algorithm that can search for a design in a few seconds without the use of any training data or hand-crafted templates.

DG3: Reason and refine a dashboard composition. The user often requires transparency of how an automated system achieves a result, and the provision of such transparency enhances the trust in automated systems [1]. In our context, it is imperative that the user be convinced of how a dashboard is created to satisfy the user-provided constraints. Yet, previous dashboard design systems employ black-box deep learning models and make it hard to reason why a certain design is chosen over alternatives [15, 51]. While MEDLEY [35] utilizes a relatively transparent algorithm and offers textual descriptions for its recommendations, the user is still required to inspect the recommended dashboard manually. Our system addresses this challenge by providing three types of explanations: 1) the scores of the current dashboard and the dashboards considered in the search process, making the user see how good the current one is in a contrastive manner, 2) token distributions that can help the user check if their intent was well reflected in the process, and 3) the contribution of each chart to the score.

4 Evaluating the Effectiveness of a Dashboard for Exploratory Data Analysis

As we discussed at Section 3, we provide the five evaluation measures as building blocks of expressing the user's own intent: *Specificity*, *Interestingness*, *Diversity*, *Coverage*, and *Parsimony*. These measures are derived from the survey of previous exploratory analysis systems, dashboard design systems, and design guidelines [5, 7, 14, 15, 23, 27, 35–37, 46, 49–51]. Table 2 summarizes the systems included in our survey regarding the perspectives considered in each system.

In this section, we first introduce how each measure has been utilized in previous studies. While these measures can be understood abstractly without having an actual implementation, we provide illustrative examples of their realization through our implementation in *Waltzboard* (Eq. (1)-Eq. (5)) to make the concepts more concrete.

Notations. In *Waltzboard*, we abstract a dashboard D as a set of charts, denoted as $D = \{C_i \mid i \in [1,n]\}$, where n is the number of charts in the dashboard. We also simplify a chart specification by representing it as a seven-dimensional tuple ($type$, x, y, $color$, t_x, t_y, t_{color}) where $type$ is the type of the chart, x, y, and $color$, (x, y, $color$) $\in A^3$ are the attributes mapped to in each axis of the chart respectively, chosen from the attributes in the given dataset, denoted as A, and t_x, t_y, and t_{color} are optional transformations applied to each axis. For example, a bar chart that represents the sum of US Gross by Creative Type is represented as (`Bar`, `Creative Type`, `US Gross`, `null`, `null`, `Sum`, `null`) and a heatmap of the mean of Worldwide Gross by Major Genre and MPAA Rating is represented as (`Heatmap`, `Major Genre`, `MPAA Rating`, `Worldwide Gross`, `null`, `null`, `Mean`). Table 1 summarizes the possible values for *type* and transformations.

Chart Types (*type*)	`Bar`, `Line`, `Pie`, `Heatmap`, `Scatterplot`, `Boxplot`, `Stripplot`
Transformations (t_x, t_y, and t_{color})	`Count`, `Mean`, `Max`, `Min`, `Sum`, `Bin`, `Time Unit` (`Year`, `Month`, `Day of week`)

Table 1: The possible values for chart type and transformation tokens.

Note that our simplified representation may not be expressive enough to express all valid chart specifications as it does not support complex visual encodings nor factorize a visualization into marks and channels. There are two reasons behind this decision. First, many previous automated dashboard design systems [15, 35, 51] abstracted visualizations at a chart-type level deliberately due to their simplicity and familiarity. Second, as we adopt a more expressive representation, e.g., Vega-lite [38], the search space for automated dashboard design also expands quickly, making the search process prohibitively challenging. A detailed list of possible values for each dimension and example specifications are available in the supplementary materials. We also allow for extending available chart types or transformations by adding new tuples.

4.1 Specificity

The first measure in our framework is **Specificity**, which quantifies the degree to which a dashboard visualizes specific data aspects of the user's interest. Although addressing the users' specific goals in EDA is important [27], it is usually hard to capture their exact goals unintrusively. One simple and common way to identify the user intent is to ask them to choose the attributes they want to analyze or select the chart type they want to see, serving as a proxy for expressing their intent. One example among dashboard design systems is MEDLEY [35], which allows the user to select the attributes of interest or the type of analysis they want to conduct.

In *Waltzboard*, extending the workflow of MEDELY [35], we allow the user to specify the tokens of interest where a token can be either an attribute, a chart type, or a transformation. Note that in contrast to MEDELY, this specification acts as "soft constraints." Charts that do not include the specified tokens can still appear in a dashboard if they contribute to other scores significantly.

Specifically, the preference of the user is specified as a preference set P, which can include attributes, chart types, and transformation types, e.g., P={`Bar`, `Worldwide Gross`, `Sum`}. We compute the specificity of a dashboard D as the average proportion of tokens in P appear in the charts of D as described below:

$$Spec(D) = \frac{1}{|D|}\sum_{C \in D} \frac{|C \cap P|}{|P|} \tag{1}$$

where C is the tuple representation of an individual chart in D. For example, considering P above, a dashboard with two charts (`Bar`, `Major Genre`, `US Gross`, `null`, `null`, `Mean`, null) and (`Bar`, `Creative Type`, `US Gross`, `null`, `null`, `Sum`, `null`) receives a specificity score of 0.5 (0.33 and 0.67, respectively).

4.2 Interestingness

Interestingness measures the number of interesting patterns a chart in a dashboard reveals. Dashboards are widely used to support decision-making by conveying interesting facts about data [5, 36] and convey intriguing insights in exploratory analysis [7]. To measure the interestingness of a chart, we need to define what interesting patterns are, which is unfortunately highly subjective and context-dependent. As a workaround, prior visualization recommendation systems leverage a range of statistical metrics to quantify the interestingness. For example, Foresight [14] and Spotfire [23] compute statistical features from the data, such as outliers and correlations, subsequently ranking the charts based on the quantity of the features.

Following these prior studies, we define $Stat(C)$ as the possible statistical features that a chart C can have. For instance, a single numeric attribute can have three possible statistical features: *High Skewness*, *High Kurtosis*, and *Has Outliers*. Similarly, a pair of numeric attributes can add two more bivariate features: *Correlated* and *Has Outliers*, resulting in a total of eight features ($3 \cdot 2 = 6$ univariate features and 2 bivariate features).

We define the interestingness of a dashboard D as the average proportion of statistical features presented in each chart as follows:

$$Int(D) = \frac{1}{|D|}\sum_{C \in D} \frac{1}{|Stat(C)|} \sum_{f \in Stat(C)} Present(f, C) \tag{2}$$

where $Stat(C)$ is set of all possible statistical features that C can contain, $Present(f, C)$ is 1 if a feature f is present in C or 0, otherwise.

The current implementation has a few limitations: 1) Even if the data has a significant statistical feature, it may not be visually evident in the charts. For instance, consider a case with a high correlation between two quantitative attributes. The correlation would be perceivable if the data is shown as a scatterplot but not if attributes are encoded differently. In addition, suppose a quantitative attribute with a significant skewness. If another categorical attribute aggregates this attribute, the skewness may not be evident in the chart. 2) There may be charts lacking statistical significance but are still perceptually intriguing, as Anscombe's Quartet [4] demonstrates. 3) There may be charts that are not statistically nor perceptually interesting, but they can become interesting if they deviate from facts or hypotheses [10]. While the above limitations are important in real-world data exploration, their detailed implementations are beyond our scope, and we left them for future research.

4.3 Diversity

Diversity measures how distinct the charts in a dashboard are. It is known that by using diverse multi-view visualization, the user can overview various perspectives of data without requiring additional cognition loads compared to single-view visualization [46]. The user study result of Voyager [49] also implies diversity in attributes and transformation types in EDA can facilitate broader exploration. Recently, Dashbot [15] incorporated diversity into the dashboard design process using the number of chart types as a reward function.

We take a simple set-based metric that computes the proportion of unique tokens in charts as described below:

$$Div(D) = \frac{\bigcup_{C \in D} C - P}{\sum_{C \in D} |C - P|} \tag{3}$$

where $|C|$ is the number of not `null` tokens in the tuple representation of a chart C. Note that we exclude the tokens in the user preference set (P) from the calculation to avoid penalizing the multiple appearances of user-preferred tokens.

4.4 Coverage

Coverage quantifies how comprehensively a dashboard visualizes the attributes in data. Data coverage has been widely used in prior visual analytics research to evaluate how their systems facilitate the user's exploration process [6, 13, 20, 31, 44, 50, 52, 53]. In particular, Sarvghad et al. [37] embedded data coverage as a Scented Widget [48] and suggested that the user can hypothesize new analytic questions and findings based on data coverage. Coverage has also been considered an essential aspect of single-chart visualization recommendation systems. For example, Voyager 2 [50] displayed additional charts by adding more attributes to the user-selected attributes to complement the coverage.

The coverage of a dashboard is quantified by the ratio of the total number of attributes ($|A|$) to the number of attributes it covers ($|A_{cov}|$, where $A_{cov} = A \cap \bigcup_{C \in D} C$). To refine this measure, we adjusted the contribution of attributes that only appear in D after a transformation, i.e., A_t, by reducing their scores from 1 to $(1-\gamma)$ ($\gamma = 0.75$ by default).

$$Cov(D) = \frac{|A_{cov}| - \gamma |A_t|}{|A|} \tag{4}$$

4.5 Parsimony

Parsimony is a criterion to minimize the number of charts in a dashboard, seeking conciseness; it improves as the dashboard consists of fewer charts. It is known that controlling the number of views in multi-view visualizations is crucial to mitigate information overload and decrease the learning cost [9, 46].

To calculate the parsimony, we employ a function that 1) has a maximum of 1 when the number of charts is at the minimum and 2) gradually decays to 0 as the number increases. To this end, we adopt an exponential function with a negative exponent as follows:

$$Par(D) = \exp\left(\frac{M - |D|}{|A|}\right) \tag{5}$$

where M is the minimum number of charts in a dashboard ($M = 3$ by default). When $|D|$ is M, i.e., a dashboard with the minimum number of charts, it receives a parsimony score of 1, and the score decreases as the number of charts increases. We divided the exponent by the number of attributes ($|A|$) to scale the decay curve; as $|A|$ becomes large, i.e., more attributes in the dataset, the score decays slower, i.e., the penalty given to a large number of charts becomes less.

Finally, given a dashboard D, we define the effectiveness score of D as the weighted sum of the five measures as follows:

$$Score(D) = w_{spec} \cdot Spec(D) + w_{int} \cdot Int(D) \ldots \tag{6}$$

where $w\cdot$ is a user-provided five-dimensional weight vector. Each metric is normalized using the mean and standard deviation obtained from randomly sampled dashboards. This normalization ensures that differences in measures' distributions do not exert a large influence relative to the user-defined weights $w\cdot$.

Searching for a design that maximizes its score is non-trivial as there are complicated trade-offs between the measures, which naturally lends the searching process for optimization. For example, one can increase the coverage score by adding more charts to D at the cost of losing the parsimony score. Another example is between specificity and interestingness; there is tension on whether or not a chart that reveals an interesting pattern but does not have the attributes of interest should be included in the dashboard. We elaborate more on formulating such an optimization problem in Section 5.1.

Chart Recommendation Systems

	Spec	*Int*	*Div*	*Cov*	*Par*
SeeDB [43]	✗	✓	✗	✗	-
Voyager [49, 50]	✓	✗	✓	✓	
ForeSight [14]	✗	✓	✗	✗	
DataSite [13]	✓	✓	✗	✗	
QuickInsight [17]	✗	✓	✗	✗	
Lux [28]	✓	✓	✗	✗	
SpotLight [23]	✗	✓	✗	✗	

Dashboard Design Systems

	Spec	*Int*	*Div*	*Cov*	*Par*
VizDeck [25]	✗	✓	✗	✗	✗
DataShot [45]	✗	✓	✓	✓	✗
QualDash [18]	✓	✗	✗	✗	✗
MultiVision [51]	✓	✗	✓	✗	✓
Dashbot [15]	✗	✓	✓	✗	✓
MEDLEY [35]	✓	✓	✗	✗	✗
Waltzboard (Ours)	✓	✓	✓	✓	✓

Table 2: Comparison between ours (the last row) and previous single chart recommendation or dashboard design systems in terms of the five aspects in our dashboard evaluation framework.

5 The *Waltzboard* System

We put forward *Waltzboard* as the main contribution of this work. Built upon our evaluation framework, *Waltzboard* not only automates the dashboard design process but also allows the user to control and understand the process. *Waltzboard* consists of two key components: a dashboard search algorithm and a user interface. We start by describing the design goals that directed our design decisions.

5.1 Automated Dashboard Design

To design a dashboard automatically without precomputation (**DG2**) while keeping the algorithm interactive (**DG1**) and interpretable (**DG3**), we formulate the design process as an optimization problem.

The objective of the optimization is to find a dashboard D_{best} consisting of n charts with each chart represented as a chart tuple, C_i (i.e., the seven-dimensional tuple described in Sec. 4), which maximizes the effectiveness score given the user-provided weights $w\cdot$ and preference set P that can be tuned by the user (Fig. 2 A):

$$D_{\text{best}} = \arg\max_{D} Score(D) \tag{7}$$

We found out this optimization problem is challenging for several reasons. First, the search space for a dashboard is combinatorially large. For example, suppose there are 20 attributes in the dataset, and we want to create a dashboard of 10 bivariate charts. For a single chart, we need to choose two attributes to visualize from the 20 attributes, leading to $C(20, 2) = 190$ possibilities. The number of possible dashboards roughly sums up to $C(190, 10) \approx 1.3 \cdot 10^{16}$, which is computationally infeasible, and this number can even increase if we consider trivariate charts (choosing x, y, and z) or transformations (e.g., t_x). In addition, the score function ($Score$)

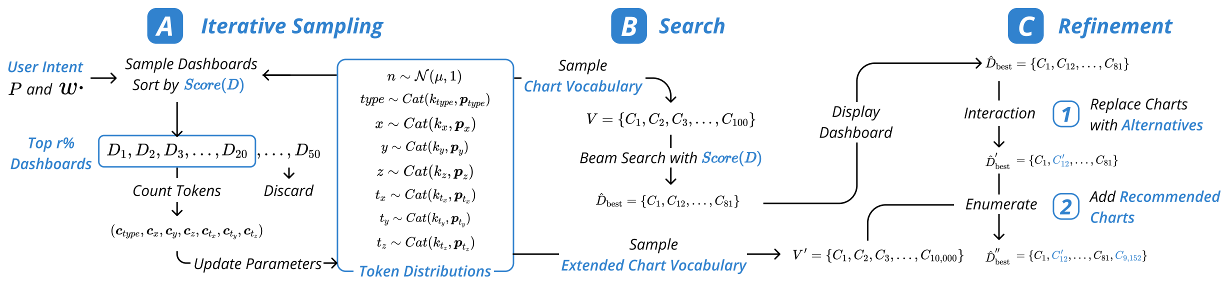
A
Iterative Sampling
User Intent
P and $w\cdot$
Sample Dashboards
Sort by $Score(D)$
Top r%
Dashboards
$D_1, D_2, D_3, \dots, D_{20}, \dots, D_{50}$
Count Tokens
Discard
$(\boldsymbol{c}_{type}, \boldsymbol{c}_x, \boldsymbol{c}_y, \boldsymbol{c}_z, \boldsymbol{c}_{t_x}, \boldsymbol{c}_{t_y}, \boldsymbol{c}_{t_z})$
Update Parameters
$n \sim \mathcal{N}(\mu, 1)$
$type \sim Cat(k_{type}, \boldsymbol{p}_{type})$
$x \sim Cat(k_x, \boldsymbol{p}_x)$
$y \sim Cat(k_y, \boldsymbol{p}_y)$
$z \sim Cat(k_z, \boldsymbol{p}_z)$
$t_x \sim Cat(k_{t_x}, \boldsymbol{p}_{t_x})$
$t_y \sim Cat(k_{t_y}, \boldsymbol{p}_{t_y})$
$t_z \sim Cat(k_{t_z}, \boldsymbol{p}_{t_z})$
Token Distributions
B
Search
Sample
Chart Vocabulary
$V = \{C_1, C_2, C_3, \dots, C_{100}\}$
Beam Search with $Score(D)$
$\hat{D}_{best} = \{C_1, C_{12}, \dots, C_{81}\}$
Sample
Extended Chart Vocabulary
$V' = \{C_1, C_2, C_3, \dots, C_{10,000}\}$
Display
Dashboard
C
Refinement
$\hat{D}_{best} = \{C_1, C_{12}, \dots, C_{81}\}$
Interaction
1
Replace Charts
with Alternatives
$\hat{D}'_{best} = \{C_1, C'_{12}, \dots, C_{81}\}$
Enumerate
2
Add Recommended
Charts
$\hat{D}''_{best} = \{C_1, C'_{12}, \dots, C_{81}, C_{9,152}\}$

Figure 3: Three phases of the search algorithm. A By iteratively generating and scoring dashboards, the probability distribution of each chart token adapts to the user-provided weights ($w\cdot$) and preference set (P). B *Waltzboard* builds *Chart Vocabulary*, consisting of $|V|$ charts drawn from the trained token distribution and searches for $\hat{D}_{\text{best}}$ using a beam search. C The user can refine the dashboard composition by 1 replacing an existing chart with a recommended alternative or 2 adding a new chart from recommendations sampled from *Extended Chart Vocabulary*.

is non-differentiable, and the search space is discrete (e.g., chart types or attributes), making classic gradient-based optimization algorithms inapplicable. As mentioned in the Section 2, recent studies leveraged deep-learning algorithms to tackle this challenge, such as reinforcement learning [15] and LSTM [51], which do not achieve our design goals.

To address these issues, we present a search algorithm that efficiently narrows down the search space. Our search algorithm consists of three phases: 1) an iterative sampling phase, 2) a search phase, and 3) a refinement phase.

5.1.1 Iterative Sampling Phase

The goal of the iterative sampling phase (Fig. 3A) is to estimate the categorical distributions of each token type so that they can be used to generate a dashboard that is effective with respect to the user intent ($w\cdot$ and P). To generate a dashboard, we first determine the number of charts in the dashboard, n, by drawing a number n from a normal distribution of variance 1, $\mathcal{N}(\mu, 1)$ and rounding it off. Then, we generate n chart tuples by sampling a value for each dimension from independent categorical distributions. For example, to determine the chart type (*type*), we sample a categorical distribution $\boldsymbol{p}$ from Dirichlet prior distribution $p(\boldsymbol{p} \mid \boldsymbol{\alpha})$, where $\boldsymbol{\alpha}$ is the Dirichlet parameter vector which is set uniform initially, $\boldsymbol{p}$ is a K-dimensional vector representing chart probabilities, and K is the number of chart types we support. The other values in the chart tuple are determined in the same way but by using separate distributions. Note that there can be interaction between dimensions. For example, if $type =$ `Scatterplot`, x needs to be a quantitative attribute. To sample only a quantitative attribute, we mask out the probabilities for categorical attributes in $\boldsymbol{p}$ to 0. By repeating this process, we build N random dashboards ($N = 50$ by default). The default value was found based on a hyperparameter optimization process detailed in the supplementary material.

As the initial probabilities for chart types and attributes are uniform, the generated dashboards are rather random and not aligned with the user intent well. To update $p(\boldsymbol{p} \mid \boldsymbol{\alpha})$ of each distribution, we evaluate each dashboard by applying the score function *Score* and pick the top r percent (20 by default) of the dashboards with the highest scores. We then update the Dirichlet parameters ($\boldsymbol{\alpha}$) of each distribution regarding the top dashboards as new observations. For example, as the Dirichlet distribution is the conjugate prior distribution of a categorical distribution, we update $\boldsymbol{\alpha}$ to the posterior distribution estimated after observing new observations (i.e., the top dashboards), which is done in practice by accumulating counts for each category. Similarly, we update μ, which is the mean of the normal distribution determining the number of charts in the dashboard. More details can be found in the supplementary materials. Note that we rerun this phase only when the user intent has changed ($w\cdot$ and P). Otherwise, the parameters fine-tuned from the previous iterative sampling phase are reused.

5.1.2 Search Phase

In the search phase, using the parameters learned from the iterative sampling phase, we search a dashboard, $\hat{D}_{\text{best}}$, which is the estimate of D_{best}, in interactive time. However, as we mentioned before, considering all possible charts is computationally infeasible. To narrow the search space, we build *Chart Vocabulary* (V) by drawing $|V|$ charts from the distributions learned in the iterative sampling phase.

$$D_{\text{best}} \approx \hat{D}_{\text{best}} = \underset{D \subset V}{\arg\max}\, Score(D) \cdot e^{-(|D| - \hat{\mu})^2} \tag{8}$$

We then perform a beam search to find a dashboard $\hat{D}_{\text{best}}$ ordering candidates based on their effectiveness score ($Score(D)$) and size ($e^{-(|D|-\hat{\mu})^2}$) where $\hat{\mu}$ is the mean number of charts in a dashboard learned in the previous phase. During the beam search, we recursively append a chart from V to the dashboard in each beam and compute the fitness. We stop the process if the whole leaf nodes' scores are less than the parent nodes' scores. The size regularization term was added to mitigate an early stopping problem where the search is stopped early due to high parsimony scores, especially when there are a minimum number of charts (i.e., 3). We set the size of the chart vocabulary ($|V|$) and the number of beams (n_{beams}) to 100 and 10, respectively.

5.1.3 Refinement Phase

The refinement phase (Fig. 3C) is a collaborative process, as we exemplified in the usage scenario (Fig. 2C). In addition to manually adding and deleting a chart, the user can collaborate with the search algorithm to obtain recommendations for existing or new charts. In this phase, the user can refine the dashboard with respect to their own perceptual and empirical rationale, e.g., adding a chart that is perceptually interesting but not included in the dashboard.

Recommending chart alternatives. The user can explore design alternatives of a chart in the dashboard and replace it if needed (Fig. 3 1); for example, the user can replace a chart with one with a different visual encoding. In the Alternatives View (Fig. 2 7), we show neighbor charts, which are the charts having an edit distance of 1 [29] to the target chart in terms of our tuple notation. For instance, if the target chart is a bar chart showing the mean of `US Gross` over `Release Date`, the neighbor charts include a line chart showing the same attributes and a bar chart showing the mean of `Production Budget` instead of `US Gross`. The neighbor charts appear sorted according to the effectiveness score acquired when the target is replaced with each.

Recommending a new chart. The user can increase the size of the dashboard by explicitly appending a new chart (Fig. 3 2). In this case, we consider the candidates in an extended vocabulary (V'), which is built by sampling more charts ($|V'| = 10{,}000$) from the trained distributions. Similar to chart alternatives, the top candidates that increase the effectiveness score of the dashboard when added appear in the Recommendations View (Fig. 2 9).

5.2 Interface

Waltzboard's interface consists of three horizontally juxtaposed panels: (1) an Intent Panel (Fig. 1 A), (2) a Dashboard Panel (Fig. 1 B), and (3) a Reasoning Panel (Fig. 1 C).

5.2.1 Intent Panel

With **Intent Panel** (Fig. 1 A), the user can express their analytic intent (DG1). It consists of two controllers, Weight Controller and Token Controller, which can be used to determine $w\cdot$ and P, respectively. The **Weight Controller** (Fig. 1 1) provides five sliders with each controlling the weight to the corresponding effectiveness measures ($w\cdot$). The user can set the weight of each measure to one of three values, 0, 1, and 2 (1 by default); a weight of 0 means that the measure is not considered in the search process, while a weight of 2 means that the measure is given double importance. For example, the user can set the weight for the specificity measure to 2 while keeping the weights for others to 1 to include more charts that contain preferred attributes. Or, the user can assign 0 to all weights except for coverage to obtain a dashboard with the highest coverage score. We chose to use three discrete weight values instead of continuous values for simplicity, reducing cognitive load while allowing for flexible prioritization. However, one may consider implementing finer-grained control to accommodate more nuanced user preferences without significantly increasing complexity.

The **Token Controller** (Fig. 1 2) provides three token lists with each enumerating the possible tokens for chart types ($type$), attributes (x, y, and z), and transformations (t_x, t_y, and t_{color}). By default, all tokens are considered in the search process, but the user can exclude a token by deselecting a checkbox (✓), which sets its probability to 0 or add a token to the user preference set P by clicking on a heart icon (♥). After expressing intent, the user can click on a Design Dashboard button to initiate the search process.

5.2.2 Dashboard Panel

The **Dashboard Panel** (Fig. 1 C) renders the result of the search process ($\hat{D}_{\text{best}}$). Each chart tuple in $\hat{D}_{\text{best}}$ is rendered as a Vega-Lite [38]. The user can pin a chart by clicking on its pin button (☆ $\rightarrow$ ★), and the pinned charts are kept in the dashboard even though the user requests new designs. We also highlight the preferred tokens (P) in chart titles, e.g., **US Gross**, to show the charts contributing to the specificity measure.

The user can further interact with each chart by clicking on a "Show Details" button (Fig. 1 3) and opening up three extra views (DG3): a Contribution View, a Statistics View, and an Alternatives View. Supporting a simple what-if analysis, the **Contribution View** (Fig. 1 4) shows the contribution of a chart to each of the five scores by showing the changes in the scores if the chart is removed. The **Statistics View** (Fig. 1 5) lists the statistical facts that contribute to the interestingness score. Finally, the **Alternatives View** (Fig. 1 6) shows alternatives for a certain chart, i.e., neighbor charts (Sec. 5.1.3), to support replacement.

The user can open a **New Chart View** (Fig. 1 7) at the bottom of the Dashboard Panel to either configure a new chart manually or choose a chart from a list of recommendations (Sec. 5.1.3).

5.2.3 Reasoning Panel

The **Reasoning Panel** (Fig. 1 C) is designed to help the user better understand the search process (DG3). While the Contribution View and Insight View support chart-level reasoning, the Reasoning Panel enables inspection on a dashboard. It consists of two views: a **Score Distribution View** (Fig. 1 8) and a **Token Probability List** (Fig. 1 9). The Score Distribution View allows the user to judge the quality of the dashboard $\hat{D}_{\text{best}}$ by comparing its scores with the ones found in the iterative sampling phase. To this end, we employ a radar chart displaying the first and third quantiles of the score distributions of sampled dashboards as shaded bands while showing the score of the current dashboard as a blue polygon. The Token Probability List illustrates the probability given to each token ($\boldsymbol{p}$) estimated in the iterative sampling phase, allowing the user to see if the probabilities correspond to their expectation; for example, a token added to the user preference set P may be received a higher probability.

Dataset	Columns	Rows
Movies [2]	10 (4 C, 5 N, 1 T, 5 E)	3,201
Birdstrikes [2]	10 (4 C, 5 N, 1 T, 5 E)	10,000
US Census [26]	12 (6 C, 6 N, 4 E)	32,561
Student Performance [12]	33 (15 C, 18 N)	649

Table 3: The datasets used in the benchmark. C, N, T, and E mean categorical, numeric, temporal, and excluded attributes, respectively.

6 PERFORMANCE BENCHMARK

In this section, we conduct a benchmark to assess the computation time of our recommendation algorithm and the impact of each phase.

Methods. We compared four search methods. In the **Random** method, we randomly generated a dashboard by first sampling the size of the dashboard from a normal distribution $\mathcal{N}(|A|, 1)$, where $|A|$ is the number of attributes in the dataset, and then sampling chart tuples from uniform categorical distributions. This method served as a baseline. In the **Iterative Sampling (I)** method, we performed only the first phase of our algorithm (Fig. 3 A) and generated a dashboard by sampling chart tuples from the trained distributions, including a normal distribution for determining the number of charts. In the **Iterative Sampling and Search (IS)** method, we performed only the first and second phases of our algorithm (Fig. 3 A and B) where a dashboard was generated through beam search on the chart tuples sampled from the trained distributions. Finally, in the **Iterative Sampling, Search, and Refinement (ISR)** method, we performed all three phases (Fig. 3 A-C), which was the setting used in *Waltzboard*. As the refinement phase requires user intervention, we assumed a simplified scenario where the user performs two refinement operations: 1) replacing the chart that least contributes to the total score with an alternative with the highest score and 2) adding one most recommended chart to the dashboard.

Datasets. In addition to the Movies dataset that was used to demonstrate a previous reinforcement learning-based model [15], we used three more datasets that had more columns or rows. We excluded categorical attributes with more than ten unique values, e.g., movie names, as they could not be effectively visualized without additional filtering or aggregation operations (shown as E attributes in Tab. 3).

Measures and Settings. For each method-dataset pair, we generated a dashboard 100 times and measured the total effectiveness score (Eq. (6)) and the completion time. Each run was independent of the others. We set the weights of scores ($w\cdot$) to 1 and generated the user preference set (P) by randomly selecting one to three tokens from possible tokens, which reflected the user behavior we observed in a user study. The benchmark was conducted on a desktop with an M1 Max CPU, 32 GB of RAM, and a Python 3.11.5 runtime.

Results and Discussion. Figure 4 shows the result of the benchmark. We could observe that our design algorithm (ISR) outperformed the baseline (Random) by 37% on average in terms of effectiveness scores. Each phase of the algorithm improved the effectiveness score to some extent on average, but a significant improvement was made during the search phase (I vs. IS). We also discovered that the refinement phase did not always improve the score and sometimes

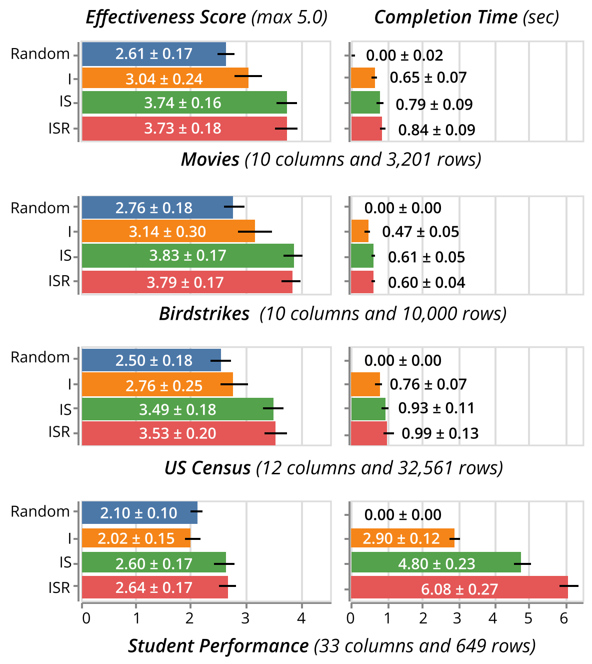

Figure 4: The result of the performance benchmark. The mean and ± 1 standard deviation of measures are shown as the length of a bar and an overlaid stroke, respectively.

degraded it (IS vs. ISR). This may be due to the simplicity of our heuristics used to simulate the user's behavior; in practical scenarios, replacing or adding a chart requires meticulous investigation.

Except for the Student Performance dataset, our algorithm completed the generation process within one second, a time limit known as the user's flow of thought to stay uninterrupted [34], demonstrating its usefulness in interactive data exploration. Even the Student Performance dataset finished within ten seconds, showing the potential to keep the user's attention on a dialogue.

The Student Performance (SP) dataset exemplifies the challenges that arise when designing a dashboard for a large number of attributes. Notably, our algorithm exhibited significantly longer execution times for the SP dataset compared to the other datasets. This can be attributed to the substantially greater number of possible chart tuples, with 1,058 tuples for the Movies dataset and 20,832 tuples for the SP dataset. In addition to responsiveness, the dashboards created for the SP dataset also displayed lower effectiveness scores than those for the other datasets. This observation suggests that the trade-off between the five performance measures intensifies when dealing with a larger number of attributes.

7 User Study

We conducted a user study to evaluate the usefulness of *Waltzboard*. We had two primary questions to answer: (1) whether the *Waltzboard*-generated dashboards can facilitate the EDA process and (2) to what extent our automated dashboard design algorithm can satisfy one's intent during exploration.

7.1 Participants and Setup

We recruited eight participants (P1-P8), including five females and three males. Most of our participants were associated with academic institutions, comprising three undergraduates and four graduates. We also had one participant who worked as a software engineer. Their educational backgrounds spanned diverse fields such as computer science, information science, and economics. We conducted a screening process to ensure they had prior experience with at least one visualization tool (e.g., Tableau, PowerBI, Excel) and one programming-based visualization library (e.g., Matplotlib). On a 5-point Likert scale ranging from 1 (never heard before) to 5 (very confident), participants self-rated their confidence levels at an average of 3.34 ($\sigma = 0.74$) for visualization tools and 3.50 ($\sigma = 1.07$) for libraries. All study sessions were conducted remotely using Google Meet, affording participants the flexibility to choose their preferred location for participation. While the number of participants is smaller than in prior studies [35, 50], future work would complement this limitation with larger and more diverse participants. Participants were compensated 22 USD for a one-hour session. With their consent, we recorded their screens and collected interaction logs for subsequent analysis.

7.2 Procedure

Inspired by the designs of previous user experiments [7, 35, 52], we used a design consisting of four blocks: tutorial, focused analysis, open-ended analysis, and post-questionnaires. In the **tutorial block** ($<$ 20 mins), participants first watched a six-minute tutorial video where we introduced the five measures of the dashboard effectiveness and the *Waltzboard*'s interface. Then, participants used our system for ten minutes to explore the Birdstrikes dataset with the following prompt: *"Suppose you are an airport employee. Please design a dashboard that describes the characteristics of birdstrikes that occur at night."* During the tutorial block, participants were allowed to ask for assistance. Next, in the **focused analysis block** ($\sim$ 15 mins), participants were asked to design a dashboard for the Movies dataset, answering a more focused prompt (**T1**): *"Suppose that you're an employee at a film company. Please design a dashboard about movies that can bring profits to determine your next investment."* In the **open-ended analysis block** ($\sim$ 15 mins), participants were tasked to explore and design a dashboard for the Student Performance dataset freely (**T2**). Note that every dataset we used in the user study is the same dataset we used in the performance benchmark described in Table 3. The study concluded with **post-questionnaires**, featuring a System Usability Scale (SUS) [8] survey and a semi-structured interview to gather feedback on system experience and address observations.

7.3 Discussions and Limitations

Benefit and diversity of intent expression. We found out the participants actively used the Intent Panel to express their intent even before they had an initial dashboard design; for example, in T1, six participants (P2-6, P8) and in T2, five participants (P1-2, P5-6, P8) expressed their intent even before they see the first look at the dataset. After obtaining an initial dashboard, they used the Intent Panel again to refine their intent. Regarding such an interaction, P7 said *"After I decided on my own analytic direction, I decreased the parsimony weight to get as much information as possible from each chart."* Moreover, P3 mentioned *"Actually, in the beginning, I don't think I really knew which chart I wanted to see, so I changed the weights and thought I was looking for a dashboard with more coverage,"* agreeing on the benefits of flexible intent expression.

Figure 5 illustrates a notable diversity in user intent specification among participants even when addressing the same task. Specifically, when we asked the participants a relatively narrow question (T1), only two participants (P6, P7) generated a dashboard by setting the weight to coverage to 0, and four participants (P1, P3-5) even increased the weight to 2, mentioning that they did not want to miss the relationship between other attributes since it was their first time to see the dataset. Participants also configured the Intent Panel in diverse ways depicted in Fig. 5 including changing weights (Fig. 5W), modifying P through clicking on the heart buttons (♥, Fig. 5U), and the exclusion of undesired attributes by unselecting the checkboxes (✓, Fig. 5E).

These observations suggest that the user exhibits diverse analytic needs in answering even focused questions, necessitating a

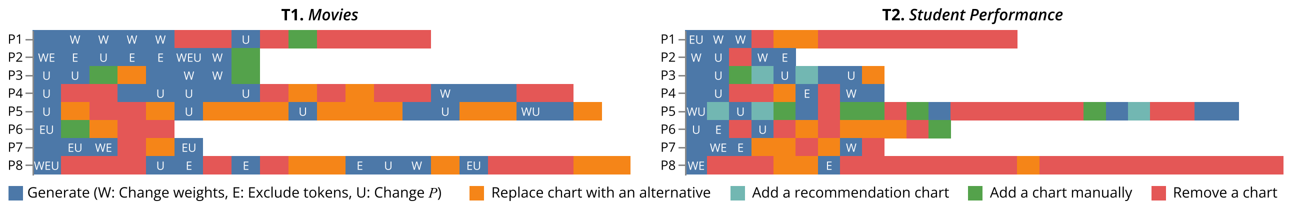

Figure 5: Event sequences depicting the user behavior in the user study.

visualization system that supports this variety.

***Waltzboard* facilitates data exploration.** We also found out the potential of our dashboard-based exploration system in promoting EDA. In a focused task (T1), our dashboards could reveal the data aspects that the user might not have initially considered. P8 mentioned *"Initially, I hadn't thought about analyzing gross by month, but as I explored the dashboard, I realized that doing so makes sense."* In an open-ended task (T2), we observed that three participants (P3-4, P7) started their analysis without expressing specific intent and decided their next steps while analyzing the initial dashboard. P3 commented *"If there are no specific questions to answer, I used to start exploring data by just looking at numbers. However, because Waltzboard shows multiple intriguing charts considering aspects like interestingness or specificity, I could start the analysis from there and tried to generate charts related to that [...]."* Indeed, we could confirm that in Figure 5-**T2**, P3 actually started the exploration without expressing any intent (Fig. 5🟦), but he changed the preference set *P* based on findings of a previous dashboard. Similarly, P4 mentioned *"I liked that each time I press the [Design Dashboard] button, I could instantly get different dashboards and pick one among them"*, emphasizing the importance of instant generation.

We also observed that participants actively refined their dashboards by interacting with the search algorithm (Fig. 5🟧, ▨, and 🟩). P7 mentioned *"When I anticipated discovering new insights based on the previous finding in the dashboard, I opened the [Alternatives View] to examine a variety of alternative charts in search of intriguing findings."* Likewise, P1 remarked *"Various charts available in the [Alternatives View] and the [Recommendations View] serve as useful guides for subsequent exploration steps."* These observations imply that our instant design of dashboards and refinement interactions could help the user start and direct data exploration.

Interpretability can help build trust when it remains succinct. After the experiment, we solicited feedback on the usefulness of visual and textual components we designed for interpretability (DG3). Overall, the participants stated that they were useful in ensuring the system was working as expected. For example, four participants (P2-4 and P6) commented that the Score Distribution View was helpful in ensuring the dashboard generated by the system was well-balanced between scores and not biased towards a particular measure, and they adjusted the weights to make it less biased. Similarly, P7 stated that he liked the Token Probability List since he could be confident about the automation happening in the internal algorithm, and it made him feel he was "controlling the system correctly."

However, we also found that several too-specific explanations are less effective to users, sometimes overwhelming them. For example, the details for a single chart (e.g., the Statistics View) are less effective for most participants; P8 said that *"While I could visually check if other measures are working well, I was not sure about interestingness because I cannot check all statistics of a chart."*. This finding suggests that the components designed for interpretability may help the user build trust, but at the same time, they can impose extra cognitive loads.

Scalability of the dashboard needs to be addressed. While the primary objective of *Waltzboard* was to enable EDA by providing a comprehensive view of all attributes in the dataset, we observed significant visual scalability challenges when handling high-dimensional data. This can be observed in the behavioral differences between T1 and T2, as shown in Figure 5. In T1, dashboards were created with an average of 9.28 charts, and during this process, an average of 3.5 charts were removed (Fig. 5🟥). On the other hand, in T2, dashboards were created with an average of 19.88 charts, and an average of 6.5 charts were removed, which is higher than in T1. This is because of the visual scalability, in which participants felt there were too many charts in the dashboard; P8 specifically mentioned that: *"The dashboard contains too many charts with unnecessary attributes."* This cannot be easily addressed with the current goal of designing a single dashboard that covers all attributes simultaneously. Under this goal, it is inevitable that the number of charts will increase according to the number of attributes, which can demand a high cognitive load for the user. We believe this visual scalability issue can be mitigated, e.g., by considering attributes relevant to the user's needs only. However, further efforts are required to enhance the overall scalability of dashboards.

Needs of effective onboarding. In the user study, it was revealed that participants could have difficulties in understanding and recalling the effectiveness measures, especially for novices. This can be seen more specifically in the responses of P3 and P7. Although we provided the tutorial video about the effectiveness framework, they found the concept of effectiveness weights abstract and difficult to grasp. However, as they compared results generated through different weight manipulations, they better understood weights and realized that their previously ambiguous intents could be expressed through criteria. This demonstrates that criteria can serve as a means of expressing intent. On the other hand, it also implies that understanding the role of the effectiveness framework requires an iterative process of multiple generations and comparisons. This difficulty in onboarding is also shown in the SUS evaluation result. *Waltzboard* rated a SUS score of 68.75, which is over average but not very high. This complexity of understanding the concepts of effectiveness framework implies effective onboarding strategies to improve the system design for better learnability and memorability [16].

8 Conclusion

We present our endeavor to facilitate the fast and automated generation of analytic dashboards reflecting the user intent. Despite the potential benefits of dashboards for exploratory data analysis, prior systems fall short in dealing with a wide range of user intent and long precomputation time. To address these issues, we first present a framework that evaluates the effectiveness of a dashboard built upon a survey of prior studies. We then present *Waltzboard*, consisting of a search algorithm and a user interface that allows the user to interact with the algorithm flexibly. Our performance benchmark shows that *Waltzboard* instantly designs an effective dashboard without long precomputation, and we could confirm that *Waltzboard* facilitates data exploration with diverse user intent through a user study.

References

[1] Microsoft hax toolkit, May 2023. 3

[2] Vega datasets. https://vega.github.io/vega-datasets, 2024. Accessed: 2024-11-18. 7

[3] S. Alspaugh, N. Zokaei, A. Liu, C. Jin, and M. A. Hearst. Futzing and moseying: Interviews with professional data analysts on exploration practices. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):22–31, 2019. doi: 10.1109/TVCG.2018.2865040 1, 2, 3

[4] F. J. Anscombe. Graphs in statistical analysis. *The american statistician*, 27(1):17–21, 1973. 4

[5] B. Bach, E. Freeman, A. Abdul-Rahman, C. Turkay, S. Khan, Y. Fan, and M. Chen. Dashboard design patterns. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):342–352, 2022. 1, 2, 3, 4

[6] S. K. Badam, Z. Zeng, E. Wall, A. Endert, and N. Elmqvist. Supporting team-first visual analytics through group activity representations. In *Graphics Interface*, pp. 208–213, 2017. 5

[7] L. Battle and J. Heer. Characterizing exploratory visual analysis: A literature review and evaluation of analytic provenance in tableau. In *Computer Graphics Forum*, vol. 38, pp. 145–159. Wiley Online Library, 2019. 1, 2, 3, 4, 8

[8] J. Brooke. Sus: a "quick and dirty'usability. *Usability evaluation in industry*, 189(3):189–194, 1996. 8

[9] 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*, 27(2):1514–1524, 2021. doi: 10.1109/TVCG.2020.3030338 5

[10] I. K. Choi, T. Childers, N. K. Raveendranath, S. Mishra, K. Harris, and K. Reda. Concept-driven visual analytics: An exploratory study of model- and hypothesis-based reasoning with visualizations. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–14. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300298 4

[11] J. Choi and J. Jo. Intentable: A mixed-initiative system for intent-based chart captioning. In *2022 IEEE Visualization and Visual Analytics (VIS)*, pp. 40–44, 2022. doi: 10.1109/VIS54862.2022.00017 2

[12] P. Cortez. Student Performance. UCI Machine Learning Repository, 2008. DOI: https://doi.org/10.24432/C5TG7T. 7

[13] Z. Cui, S. K. Badam, M. A. Yalçin, and N. Elmqvist. Datasite: Proactive visual data exploration with computation of insight-based recommendations. *Information Visualization*, 18(2):251–267, 2019. doi: 10.1177/1473871618806555 5

[14] C. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Recommending visual insights. *Proceedings of the VLDB Endowment*, 10(12), 2017. 1, 2, 4, 5

[15] D. Deng, A. Wu, H. Qu, and Y. Wu. Dashbot: Insight-driven dashboard generation based on deep reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):690–700, 2022. 1, 2, 3, 4, 5, 6, 7

[16] V. Dhanoa, A. Hinterreiter, V. Fediuk, N. Elmqvist, E. Gröller, and M. Streit. D-tour: Semi-automatic generation of interactive guided tours for visualization dashboard onboarding. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–11, 2024. doi: 10.1109/TVCG.2024.3456347 9

[17] 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 International Conference on Management of Data*, SIGMOD '19, p. 317–332. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3299869.3314037 5

[18] M. Elshehaly, R. Randell, M. Brehmer, L. McVey, N. Alvarado, C. P. Gale, and R. A. Ruddle. Qualdash: Adaptable generation of visualisation dashboards for healthcare quality improvement. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):689–699, 2021. doi: 10.1109/TVCG.2020.3030424 1, 5

[19] W. Epperson, D. Jung-Lin Lee, L. Wang, K. Agarwal, A. G. Parameswaran, D. Moritz, and A. Perer. Leveraging analysis history for improved in situ visualization recommendation. *Computer Graphics Forum*, 41(3):145–155, 2022. doi: 10.1111/cgf.14529 2

[20] M. Feng, E. Peck, and L. Harrison. Patterns and pace: Quantifying diverse exploration behavior with visualizations on the web. *IEEE transactions on visualization and computer graphics*, 25(1):501–511, Sept. 2018. doi: 10.1109/TVCG.2018.2865117 5

[21] S. Few. *Information dashboard design: The effective visual communication of data.* O'Reilly Media, Inc., 2006. 1

[22] D. Gotz and Z. Wen. Behavior-driven visualization recommendation. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, p. 315–324. Association for Computing Machinery, New York, NY, USA, 2009. doi: 10.1145/1502650.1502695 2

[23] C. Harris, R. A. Rossi, S. Malik, J. Hoffswell, F. Du, T. Y. Lee, E. Koh, and H. Zhao. Insight-centric visualization recommendation. *arXiv preprint arXiv:2103.11297*, 2021. 1, 4, 5

[24] 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*, CHI '19, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300358 2

[25] A. Key, B. Howe, D. Perry, and C. Aragon. Vizdeck: Self-organizing dashboards for visual analytics. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, p. 681–684. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2213836.2213931 2, 5

[26] R. Kohavi. Census Income. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5GP7S. 7

[27] H. Lam, M. Tory, and T. Munzner. Bridging from goals to tasks with design study analysis reports. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):435–445, 2018. doi: 10.1109/TVCG.2017.2744319 1, 4

[28] D. J.-L. Lee, D. Tang, K. Agarwal, T. Boonmark, C. Chen, J. Kang, U. Mukhopadhyay, J. Song, M. Yong, M. A. Hearst, and A. G. Parameswaran. Lux: Always-on visualization recommendations for exploratory dataframe workflows. *Proc. VLDB Endow.*, 15(3):727–738, nov 2021. doi: 10.14778/3494124.3494151 5

[29] V. I. Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, vol. 10, pp. 707–710. Soviet Union, 1966. 6

[30] Y. Lin, H. Li, A. Wu, Y. Wang, and H. Qu. Dashboard design mining and recommendation. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–15, 2023. doi: 10.1109/TVCG.2023.3251344 2

[31] Z. Liu and J. Heer. The effects of interactive latency on exploratory visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2122–2131, 2014. doi: 10.1109/TVCG.2014.2346452 3, 5

[32] R. Ma, H. Mei, H. Guan, W. Huang, F. Zhang, C. Xin, W. Dai, X. Wen, and W. Chen. Ladv: Deep learning assisted authoring of dashboard visualizations from images and sketches. *IEEE Transactions on Visualization and Computer Graphics*, 27(9):3717–3732, 2021. doi: 10.1109/TVCG.2020.2980227 2

[33] 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*, 25(1):438–448, 2019. doi: 10.1109/TVCG.2018.2865240 2

[34] J. Nielsen. *Usability engineering*. Morgan Kaufmann, 1994. 8

[35] A. Pandey, A. Srinivasan, and V. Setlur. Medley: Intent-based recommendations to support dashboard composition. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1135–1145, 2023. doi: 10.1109/TVCG.2022.3209421 1, 2, 3, 4, 5, 8

[36] 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*, 25(1):682–692, 2018. 1, 3, 4

[37] A. Sarvghad, M. Tory, and N. Mahyar. Visualizing dimension coverage to support exploratory analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):21–30, 2016. 1, 4, 5

[38] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, 2017. doi: 10.1109/TVCG.2016.2599030 4, 7

[39] E. Segel and J. Heer. Narrative visualization: Telling stories with data.

IEEE transactions on visualization and computer graphics, 16(6):1139–1148, 2010. 2

[40] A. Srinivasan and V. Setlur. BOLT: A Natural Language Interface for Dashboard Authoring. In T. Hoellt, W. Aigner, and B. Wang, eds., *EuroVis 2023 - Short Papers*. The Eurographics Association, 2023. doi: 10.2312/evs.20231035 2

[41] B. Steichen, G. Carenini, and C. Conati. User-adaptive information visualization: using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, p. 317–328. Association for Computing Machinery, New York, NY, USA, 2013. doi: 10.1145/2449396.2449439 2

[42] M. Tory and V. Setlur. Do what i mean, not what i say! design considerations for supporting intent and context in analytical conversation. In *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 93–103, 2019. doi: 10.1109/VAST47406.2019.8986918 2

[43] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: Efficient data-driven visualization recommendations to support visual analytics. *Proc. VLDB Endow.*, 8(13):2182–2193, sep 2015. doi: 10.14778/2831360.2831371 2, 5

[44] E. Wall, L. M. Blaha, L. Franklin, and A. Endert. Warning, bias may occur: A proposed approach to detecting cognitive bias in interactive visual analytics. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 104–115. IEEE, Oct. 2017. doi: 10.1109/vast.2017.8585669 5

[45] 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*, 26(1):895–905, 2020. doi: 10.1109/TVCG.2019.2934398 2, 5

[46] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pp. 110–119, 2000. 1, 3, 4, 5

[47] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson. The what-if tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):56–65, 2020. doi: 10.1109/TVCG.2019.2934619 1

[48] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, 2007. doi: 10.1109/TVCG.2007.70589 5

[49] 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*, 22(1):649–658, 2016. doi: 10.1109/TVCG.2015.2467191 1, 2, 4, 5

[50] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, p. 2648–2659. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3025453.3025768 1, 2, 3, 4, 5, 8

[51] 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*, 28(1):162–172, 2021. 1, 2, 3, 4, 5, 6

[52] Z. Zeng, P. Moh, F. Du, J. Hoffswell, T. Y. Lee, S. Malik, E. Koh, and L. Battle. An evaluation-focused framework for visualization recommendation algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):346–356, 2021. 5, 8

[53] J. Zhao, M. Fan, and M. Feng. Chartseer: Interactive steering exploratory visual analysis with machine intelligence. *IEEE Transactions on Visualization and Computer Graphics*, 28(3):1500–1513, 2022. doi: 10.1109/TVCG.2020.3018724 5

[54] M. Zhuang, D. Concannon, and E. Manley. A framework for evaluating dashboards in healthcare. *IEEE Transactions on Visualization and Computer Graphics*, 28(4):1715–1731, 2022. doi: 10.1109/TVCG.2022.3147154 1