Visual Informatics 8 (2024) 106–114

Contents lists available at ScienceDirect

Visual Informatics

journal homepage: www.elsevier.com/locate/visinf

Research Article

AVA: An automated and AI-driven intelligent visual analytics framework[☆]

Jiazhe Wang [a,b,c], Xi Li [b], Chenlu Li [b], Di Peng [b], Arran Zeyu Wang [b], Yuhui Gu [b], Xingui Lai [b], Haifeng Zhang [b], Xinyue Xu [b], Xiaoqing Dong [b,c], Zhifeng Lin [b], Jiehui Zhou [d], Xingyu Liu [d], Wei Chen [d,*]

[a] Intelligent Big Data Visualization Lab at Tongji University, Shanghai, China
[b] Ant Group, China
[c] Alibaba Group, China
[d] State Key Lab of CAD&CG, Zhejiang University, Hangzhou, China

ARTICLE INFO

Article history:
Received 10 January 2024
Received in revised form 7 June 2024
Accepted 11 June 2024
Available online 13 June 2024

Keywords:
Automated visual analytics
Visualization recommendation
Insight mining

ABSTRACT

With the incredible growth of the scale and complexity of datasets, creating proper visualizations for users becomes more and more challenging in large datasets. Though several visualization recommendation systems have been proposed, so far, the lack of practical engineering inputs is still a major concern regarding the usage of visualization recommendations in the industry. In this paper, we proposed *AVA*, an open-sourced web-based framework for **A**utomated **V**isual **A**nalytics. AVA contains both empiric-driven and insight-driven visualization recommendation methods to meet the demands of creating aesthetic visualizations and understanding expressible insights respectively. The code is available at https://github.com/antvis/AVA.

1. Introduction

Visualization maps abstract data to visual presentations (e.g., position Gleicher et al., 2011, color Tseng et al., 2023, and shape Smart and Szafir, 2019) for users' comprehension and is a powerful way to reveal hidden insights and communicate compelling stories from data (Battle and Ottley, 2023; Szafir et al., 2023; North, 2006). However, designing visualization for effective visual data communication is often challenging, labor-intensive time-consuming, and highly dependent on expertise such as visual communication design, user experience design, and data analysis (Szafir et al., 2023; Qin et al., 2020).

Recently, with the rapid growth of data scales and the rise of artificial intelligence techniques, machine intelligence began to be incorporated into the visualization process to improve the communicative effectiveness for users (Chen et al., 2023). For Example, some work focuses on the data level, such as automatically selecting the dimension fields to be visualized within complex data (Cui et al., 2019a), recommending explainable data transformations (Wu et al., 2023a), and resulting valuable data facts or insights (Demiralp et al., 2017). For a given input of data or other modalities (Cui et al., 2019b; Yu and Silva, 2019), there are also studies that explored the effectiveness of both data and non-data visual encodings, e.g., data selection (Kaul et al., 2021), color (Tseng et al., 2024), and ensembles (Szafir et al., 2016). However, there are still challenges that need to be addressed:

- The existing visualization recommendation systems often have limited and fixed recommendation rules or training data, which are difficult to adapt to actual variable visualization requirements that may be important for real-world use cases such as various chart styles and customized designs (Qin et al., 2020).
- Although existing recommendation systems can help generate visualization charts, communicating recommended insights to target users from data still requires additional efforts, such as organizing vivid data stories and written descriptions (Shi et al., 2020).

To address these challenges, we propose *AVA*, an automated, open-sourced, and AI-driven intelligent visual analytics framework, to help developers with different experiences to make visualization efficient. We worked closely with experts from a data intelligence department in a well-established IT company and extracted requirements through iterative discussions with them.

☆ Given his role as Editor in Chief of this journal, Wei Chen had no involvement in the peer-review of this article and has no access to information regarding its peer-review. Full responsibility for the peer-review process for this article was delegated to Anan Liu.

* Corresponding author.
E-mail address: chenvis@zju.edu.cn (W. Chen).

https://doi.org/10.1016/j.visinf.2024.06.002
2468-502X/

In particular, these requirements are drawn from real-world complex visual development scenarios. This motivates us to propose *AVA*, which mainly integrates two modules, namely, the base module and the recommendation module. The base module consists of some basic syntax and rules to help with chart recommendation. The recommendation module includes the main process of data visualization, such as data pre-processing, empirical-driven recommendation, insight-driven recommendation, and narrative data interpretation. We demonstrate the effectiveness and usability of *AVA* through case studies and a comparative analysis based on real-world scenarios. The main contributions of this paper are as follows:

- An engineering framework for intelligent visual analytics with low learning and customization costs and suitable for diverse business scenarios.
- A visual recommendation pipeline that incorporates empiric- and insight-based approaches and supports narrative data interpretation.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 proposes the design requirements. Section 4 introduces the architecture of the *AVA* framework. We present two case studies and a comparative analysis to validate our work in Section 5, discuss in Section 6, and conclude in Section 7.

2. Related work

2.1. Insight extraction

Insight extraction focuses on extracting useful data facts, also known as insights, from the data (Battle and Ottley, 2023; North, 2006; Gotz and Zhou, 2009). Insights can be classified into various types, such as outstanding, dominance, top-k, outlier, increase trend, etc Lin et al. (2018), Tang et al. (2017) and Ma et al. (2021). If the amount of data is not large, all possible fields can be enumerated based on data type and name (Wongsuphasawat et al., 2015). DataSite (Cui et al., 2019a) analyzes the data through predefined algorithms, such as calculating Pearson correlation coefficients (Benesty et al., 2009) for all combinations of numerical attributes. There are also studies to evaluate insights by designing relevant metrics to identify more valuable insights. For example, Voder (Srinivasan et al., 2018) uses a set of predefined heuristics to group data facts into three tiers. Tier 1 consists of the most prominent data facts. Ding et al. (2019) designed an insight evaluation algorithm to eliminate easily inferable insights to achieve high-quality insight results. Mafrur et al. (2018) proposed a hybrid objective utility function, which captures both the importance and the diversity of insights. Vartak et al. (2015) built a system, SeeDB, a DBMS middleware, that uses a deviation-based utility metric to display large deviations from some reference.

As the dimensionality of the data increases, the space of possible insights becomes very large, making it difficult for the computational speed to meet the interaction requirements. In order to accelerate the computation, Foresight (Demiralp et al., 2017) proposed a sketch composition for fast approximate computation of insight metrics. Calliope (Shi et al., 2020) which uses a logic-oriented Monte-Carlo tree search algorithm. The algorithm avoids the time-consuming enumeration of the data space via a reward function and a logic filter to ensure the quality of the generation results. Databiting (Rey et al., 2024) supports interaction with personal data to provide enriched insight exploration in mobile devices. Recent advancements in this field include systems like XInsight (Ma et al., 2023b), which provides a general framework for explainable data analysis through the lens of causality, offering qualitative and quantitative explanations that significantly improve human understanding and confidence in data analysis outcomes. Similarly, InsightPilot (Ma et al., 2023a), an LLMs (Large Language Models)-based system, simplifies the data exploration process by issuing a sequence of analysis actions to explore the data and generate insights from natural language questions. In addition, the industry has developed systems with similar automatic insight generation features like Microsoft Power BI,[1] Google Sheets,[2] and Amazon QuickSight.[3]

Besides, novel visualization and visual analytics approaches utilizing various statistical and modeling techniques have also been proposed to guide efficient insight exploration under analytics contexts (Zhou et al., 2022a). For example, characterizing empirical and statistical features as guidance (Ceneda et al., 2016; Zhou et al., 2021; Kale et al., 2023), contextualizing data to mitigate bias and paradox (Armstrong and Wattenberg, 2014; Gotz et al., 2016; Dimara et al., 2018), aggregating datasets to support meaningful interpretations (Xiong et al., 2019; Borland et al., 2024; Wang et al., 2024), as well as analyzing insights under domain-specific expertise and usage scenarios such as supporting event sequence analysis (Gotz and Stavropoulos, 2014; Guo et al., 2017; Jin et al., 2020), privacy and fraud analysis (Zhou et al., 2022b, 2023; Nanayakkara et al., 2024), and geospatial insight exploration (Wood et al., 2007; Chen et al., 2017; Zhou et al., 2018). These methods underscore the ongoing focus of insight extraction to support visualizations.

However, despite the rich set of methods provided by the aforementioned works for insight extraction, there is a notable gap in research and front-end tooling when it comes to determining the most effective visual representations for different types of insights. Users may have different preferences across different types of insights and tasks in visualization (Quadri and Rosen, 2021; Quadri et al., 2024), current practices may not fully exploit the potential of visualization to enhance the interpretability and impact of the extracted insights. In our work, AVA takes into account the characteristics of the data itself as well as different user preferences to control the output of the insight results.

2.2. Visualization recommendation

An increased level of interest has recently emerged in facilitating visual data exploration by recommending visualizations (Zhou et al., 2022a). As early as 1986, Mackinlay (1986) proposed APT to rank perceptual channels based on the type of data fields. This provides a reference guideline for the visual encoding recommendation of data, which is employed in systems such as Tableau (Mackinlay et al., 2007) and Voyager (Wongsuphasawat et al., 2015, 2017).

Depending on whether the user explicitly specifies the purpose of the analysis or implicitly infers user intent from the data, the type of chart that is appropriate for that type of task can be recommended (Gotz and Zhou, 2009; Zeng et al., 2021). For example, in DataVizard (Ananthanarayanan et al., 2018), a line chart is recommended when the user wants to see the trend of the data, while when the user selects a numeric variable and a category variable, it can be inferred that the user wants to compare the number of different categories, which is where a bar chart is recommended. In addition to those based on empirical rules, prior studies recommend visualizations based on exploring users' intents based on their behaviors or interactions (Gotz and Wen, 2009; Brown et al., 2014) as well as constraint-based optimization

[1] https://www.microsoft.com/en-us/power-platform/products/power-bi/.

[2] https://www.google.com/sheets/about/.

[3] https://aws.amazon.com/quicksight/.

using off-the-shelf solvers for visual design criteria (Moritz et al., 2018; Lin et al., 2020). Besides, Data2Vis (Dibia and Demiralp, 2019), VizML (Hu et al., 2019), as well as DataShot (Wang et al., 2019) employ end-to-end models to directly learn the mapped relationships between data and encoding. Chen et al. (2021), on the other hand, provides VizLinter, an architecture that includes a linter and fixer to check existing visualizations for violations of design guidelines and provide suggestions for fixes using linear programming.

In addition, prior work investigated how to recommend visual encodings that involve aesthetics and perception. For example, LADV (Ma et al., 2020) formalized several aesthetic metrics to optimize the grid layout of the dashboard. MobileVisFixer (Wu et al., 2020) employed a reinforcement learning framework to adjust layout parameters to generate more mobile-friendly visualizations. ColorCrafting (Smart et al., 2019) employed an algorithmic approach that models designer practices by analyzing patterns in the structure of designer-crafted color ramps.

More recently, with the widespread development of machine learning models, AI-powered tools have been proposed to support visualization recommendations as well (Chen et al., 2023). Calliope (Shi et al., 2020) and Autoclips (Shi et al., 2021) supported automated generations of data stories and videos from large datasets. Viz2viz (Wu et al., 2023b) utilized diffusion models to support generating aesthetic stylized visualizations. ChartSpark (Xiao et al., 2023) further incorporated semantic context into generated pictorial visualizations with text-to-image models. The incorporation of LLMs in visualization tasks has opened new possibilities. Li et al. Li et al. (2024) showcases the effectiveness of LLMs, specifically GPT-3.5, in generating Vega-Lite specifications from natural language descriptions. ChartGPT (Tian et al., 2024) introduced natural language interfaces to generate charts from abstract natural language inputs with LLMs support. LIDA (Dibia, 2023) is another example that combines LLMs with image generation models to produce charts and infographics. It defines a four-stage task that combines LLMs and image generation models to interpret datasets and analysis objectives. Data Formulator (Wang et al., 2023) introduces a paradigm where an AI agent separates high-level visualization intent from low-level data transformation steps, enabling the automatic generation of desired visualizations from defined data concepts. The generative methods serve as good data creativity tools when generating charts, however, they cannot fit well with the employed data itself, resulting in low performance in generating actionable insights (Basole and Major, 2024).

The aforementioned literature offers a variety of effective methods and means for automatically recommending chart types. However, in real-world engineering practice, there is often a need for interpretability and certainty in the recommendation results, as well as a user-friendly development approach on the engineering side. AVA provides empiric- driven and insight-driven visualization recommendation methods. It also offers an explainable rule system and a suite of front-end tools centered around automated visualization. These features help to ensure that the chart recommendations provided are interpretable and robust in an engineering context. Besides, existing methods still require users to interpret the insights themselves after recommending visualizations. AVA further supports narrative visualization and helps users understand the information conveyed by the visualization through descriptive text.

3. Design consideration

In this section, we provide an in-depth analysis of the actual scenario of visualization development and present our design goals.

3.1. Visualization development scenarios and issues

To better understand the problems visualization developers encountered in practice, we conducted semi-structured interviews with three experts. All three experts are from an IT company; one is a product manager, one is a front-end engineer, and the other is a designer. They all have extensive experience in developing visualization systems such as business intelligence. Typically, a product manager will come up with a requirement and write a product requirements document. This document needs to go through several rounds of reviews and iterations before it becomes a formal requirement. After that, the project is scheduled. The front-end and back-end engineers will develop based on the requirements document and the design drawings. After the development is completed, it needs to go through self-testing, front-end and back-end debugging, and regression testing, and if bugs are found in this process, they need to be fixed. Finally, it needs to be checked before it can be put online in the production environment gradually. Three common issues were summarized during the interviews.

- **I1 Lack of visualization expertise.** The lack of visualization design knowledge and development skills for developers creates difficulties in implementing data visualization requirements, especially in some innovation-incubated and urgent projects.
- **I2 Diverse requirements for different businesses.** Different businesses will have different needs, such as the configuration of the chart, and it often takes a lot of time to make personalized changes in actual development.
- **I3 Difficult to choose the right chart.** In the beginning, developers often do not know what form of visualization to use for their data, which requires many iterations to find the suitable one. In addition, differences in the data itself can affect the comprehensibility of the charts.

3.2. Design goals

After learning about the problems frequently encountered by people involved in visualization projects, we also interviewed them about what features they would like to see in existing visualization development frameworks to solve this problem. Therefore, we summarized three design goals.

- **G1 Reduce development effort.** Developers want to simplify the workload and knowledge threshold in the visualization chart development. Developers do not always want to be involved in every step of the visualization pipeline. It would ease their burden to recommend good visualizations, both individual visualizations and collections of visualizations, such as dashboards.
- **G2 Support personalized configuration.** Support custom configuration of charts to reduce tedious code adjustments. It would be beneficial to open customization capabilities, such as customizing specific recommendation rules on/off and independently modifying a part of the recommendation process.
- **G3 Adapt to different data.** Users want to be able to select a suitable chart according to different data characteristics. For example, a line chart usually reflects the trend well for time-series data, but when there are many outliers in the data, a scatter chart has a clearer presentation. On the other hand, when users are faced with unfamiliar data, a visualization generated adaptively based on the data can help users explore the data quickly.

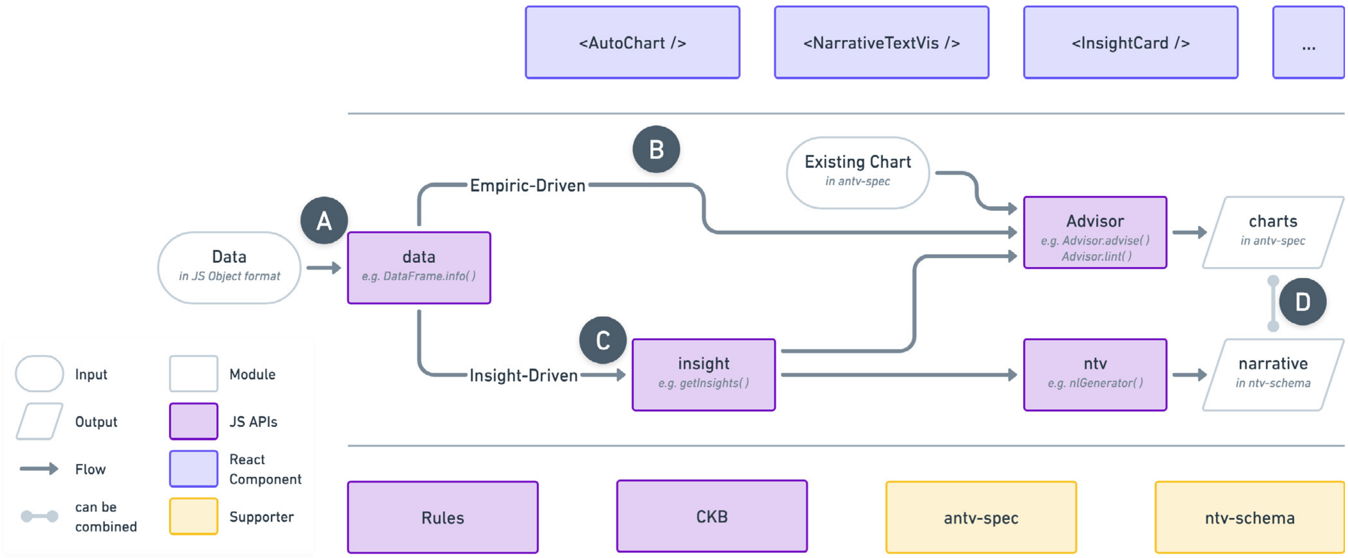

Fig. 1. The overall architecture of AVA 3.0. Multiple npm packages are integrated into one: *@antv/ava*. It keeps core features as APIs, which simplifies the usage while ensuring flexibility. This framework provides react components *@antv/ava-react* and the new *NTV* module supports narrative text visualization. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4. Architecture

In order to achieve the above goals, we propose *AVA*. The *AVA* framework contains several steps from data to visualization, as shown in Fig. 1. The framework is based on the visualization declarative syntax *antv-spec* and the chart knowledge base *CKB*. Data first goes through the data pre-processing module, which generates processed data and related statistics. Two visual recommendation paths are then available, empiric-driven recommendation and insight-driven recommendation. In addition, AVA offers module NTV for the insight-driven recommendation, which can provide a natural language description of the insight conclusion. To integrate some of these aforementioned functionalities, the framework offers a component library for extensible capability.

4.1. Bases

The bases of the framework include a declarative syntax, a chart knowledge base, the selected visualization rules, and a declarative schema to support chart recommendations.

4.1.1. antv-spec

The definition of specifications is an important part of a visualization system. There have been proposed a variety of specifications such as Vega (Satyanarayan et al., 2015) and Vega-Lite (Satyanarayan et al., 2016). But existing schemas neither can cover the necessary aspects of automated visual analytics, nor fit the rapidly changing visualization systems. Hence we also need to define a unique JSON specification based on *AVA*'s demands. Thus we proposed the *antv-spec*[4] as a declarative grammar that supports various technology stacks of *AntV*. As shown in Fig. 2, its definition consists of metadata, canvas attributes (e.g. width and height), data mapping attributes (e.g. marks), and data annotation attributes (e.g. data definition). Also, the *antv-spec* grammar can be linked to G2Plot[5] for rendering by an adaptor.

[4] https://github.com/antvis/antv-spec.

[5] https://github.com/antvis/g2plot.

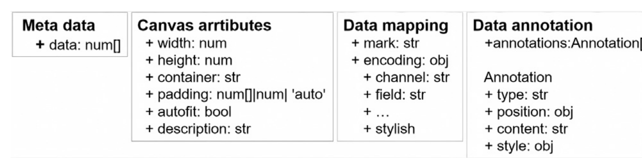

Fig. 2. *antv-spec*, a declarative grammar that supports various technology stacks of *AntV*.

4.1.2. CKB

The chart knowledge base, *CKB*, is a library that offers a knowledge base for chart wikis in a JSON format. We aim to build a standard wiki for every type of visualization. Thus users do not have to struggle with different names, aliases, or definitions of the same chart type via *CKB*. *CKB* enables users to quickly build their own chart dictionary product, for example, ChartCube.[6] Besides, with *AVA*, users will be able to easily build a chart recommendation system with *CKB* and their customized rules. The full definition of the knowledge bases can be found on Github.[7]

4.1.3. Rules

The selected visualization rules, *Rules*, is a library that offers a set of visualization rules that are important for industry usage scenarios defined in *CKB* format. There are two types of rules, hard and soft. While hard rules are mandatorily agreed on charts, and soft rules are weighted for computation. Previous works such as Voyager (Wongsuphasawat et al., 2015) and Draco (Moritz et al., 2018) both employ a lot of existing visualization rules which significantly raise the computational cost of the visualization searching space. We aim to abstract the core rules for real usage scenarios to make it more effective and reduce the computational cost. With the help of a group of designers, analysts, and product managers, we abstracted 13 hard and soft rules in *Rules* which can faithfully describe the most important cases in industry applications, such as the "data-check" rule, "data-field-qty" rule and "no-redundant-field" rule. We also define different

[6] https://chartcube.alipay.com/.

[7] https://github.com/antvis/AVA.

weights for our soft rules based on the business demands. The full definition of the rules can be found on GitHub.[8]

4.1.4. ntv-schema

In order to standardize the style of textual expression, *AVA* employs the concept of visual channel mapping from data visualization theory to regulate the visual mapping representation of text and implement it in *NTV*.

To formalize this specification, *AVA* has designed a set of declarative schemas for interpreting the text, namely *ntv-schema*. This will facilitate the circulation of the schema among various systems and also serve as preparation for future intelligent recommendations.

The *ntv-schema* is divided into two layers: the structural layer and the phrase layer. In terms of the structural layer, the whole interpretation structure is called narrative, which consists of a headline and multiple sections. Each section is composed of several paragraphs, and each paragraph consists of multiple phrases. The phrase layer reflects the most significant difference between "data-describing text" and ordinary text. Phrases are categorized into three types: text, entity, and custom. The text type is ordinary plain text. The entity type represents phrases with data meaning that map data to text, which is the main content for visualizing the interpretation of the text. The custom type is a phrase slot that allows users to customize, often used for implementing phrase-level interactions.

4.2. Recommendation pipeline

The main process is visualization recommendation, which includes data pre-processing, empiric-driven recommendation, insight-driven recommendation, and narrative data interpretation.

4.2.1. Data pre-processing

Data pre-processing (Fig. 1-A) is the data processing and pre-computing library of *AVA*. It contains general functions and dataset components. The general functions are divided into four parts: Analyzer, Statistics, Random, and Utils.

The Analyzer is used to analyze what type a value belongs to, which may be one integer, float, date, string, or null, and can be further classified into nominal, ordinal, discrete, etc. The Random is a random value generator that can be used to generate mock data as well as color palettes, etc. Some statistical information of the data is calculated by Statistics, such as mean, quartile, variance, and Pearson correlation. Utils consist of data type inference and utility data operations. To improve performance, *Data pre-processing* also uses WeakMap[9] to cache some statistics.

Dataset implements Series, a one-dimensional data structure, DataFrame, a two-dimensional data structure, and Graph, a topological relational data structure. Representing the raw data in an appropriate data structure facilitates the selection of specific visualization forms and insight extraction algorithms. The structure of *Data pre-processing* is shown as Fig. 3.

4.2.2. Empiric-driven recommendation

Advisor is the empiric-driven chart recommendation (Fig. 1-B) and lint lib of *AVA*. It employs a rule-based chart recommendation model. The pipeline of *Advisor* can be illustrated in Fig. 4. Both the recommendation and lint processes are based on empirical rules. For a given dataset, the *Advisor* component will compute the score through every type of chart among those rules, and output the most influential one. The Linter component could compute how the input chart can fit the empirical rules, then fix the errors or give feedback to the user. The computation of both components is based on the above *Rules*. The output will be a set of visualization advice that would make the chart accomplish all the hard rules and achieve the largest weight for the soft rules. The advice follows the *antv-spec* grammar which could be translated to G2Plot for rendering.

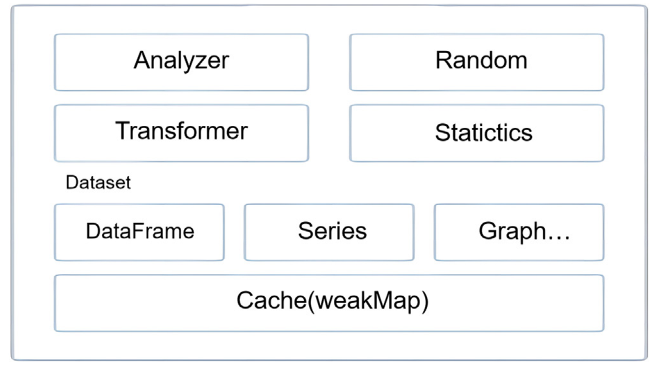

Fig. 3. The *Data pre-processing* contains the following components and functions: (i) General functions and (ii) DataSet (non-entity concept).

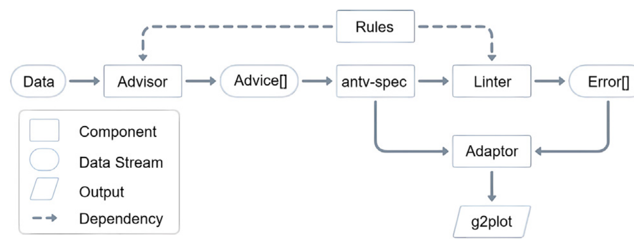

Fig. 4. The *Advisor* contains two main tool functions: (i) *Advise()*, which recommends charts automatically and (ii) *Lint()*, which provides chart optimization suggestions.

As shown in Fig. 4, for an input dataset, it would be fed into the *Advisor* component first, which would output with visualization advice. The advice would be adapted to G2Plot via *antv-spec* for rendering as a chart. Finally, the chart would go through the Linter component which could output visualization problems as an optional output for users to improve their charts.

Besides, the two components can be used together or individually. For example, users are allowed to input a manual chart into the Linter to prettify it. In Ant Group's business intelligence platform, Linter helps users at every level to create charts and get insights easily. Using smart and appropriate prompts, users no longer need to look for the chart configuration and the chart configuration will look for users automatically. When there are some problems in a chart, Linter will find problems quickly and prompt the user through the yellow breathing light. After that, the user can follow the guide and fix problems, as shown in Fig. 5.

4.2.3. Insight-driven recommendation

GetInsights (Fig. 1-C) is the insight-driven visual exploration library of *AVA*. It employs a pruning-based insight exploration model, which classifies data by insight type through subspace enumeration, pruning, and pattern matching.

Then the insight score is calculated and the final output includes the insight result, meta-data, and optional output structure. The insight score consists of numerical impact and task-related impact. The numerical impact reflects the importance of the insight for the whole data set, which is defined on a specific impact measure (e.g., as a percentage).

8 https://github.com/antvis/AVA.

9 https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/WeakMap.

Fig. 5. An instance of using *Linter* to find and fix problems in a line chart.

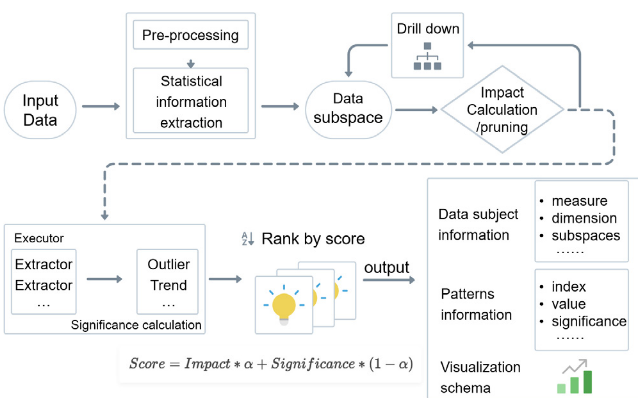

Fig. 6. The pipeline of *getInsights*, which contains the following steps: (i) Data pre-processing, (ii) Significance calculation, and (iii) Insight output.

Impact needs to be normalized and satisfy anti-monotonic conditions in order to be fairly comparable. In practice, users sometimes have interest preferences for specific data subjects and insight types, so task impact is added to adjust the score of the output.

For graph data, *getInsights* provides graph-based subspace enumeration capability, In which subspaces can exist in the form of both subgraphs and sub-tables. Insight of graph data can be expressed by annotations and layouts. The pipeline of *getInsights* can be shown in Fig. 6.

4.2.4. Narrative data interpretation

In the comprehensive presentation of the entire data analysis process, besides visual charts, the utilization of textual descriptions to depict data phenomena and provide insightful conclusions for analysis (Fig. 1-D) is of great significance. Adding textual information effectively avoids the ambiguity of visualization, and precisely presents the background information and noticeable insights to the readers. Moreover, as a visual presentation element, text can emphasize key information through kinds of visualization mapping channels, such as font color, background color, size, icons, etc., thereby enhancing the efficiency with which users obtain insights. Narrative visualization combines narrative and storytelling with data visualization to communicate information

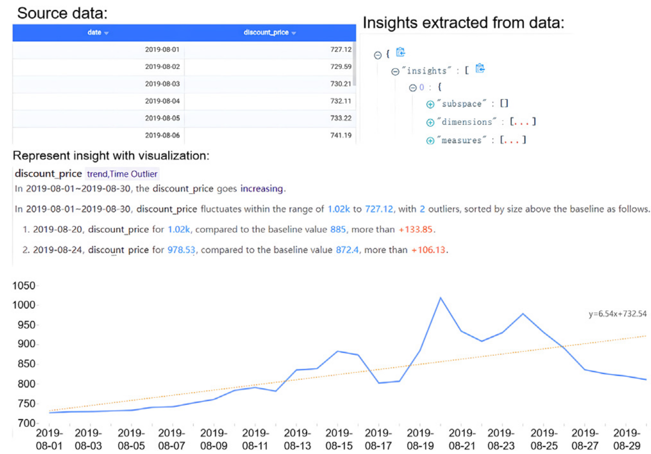

Fig. 7. An instance of insight-driven recommendation process for user P2.

effectively and expressively. It utilizes the norms of communicative and exploratory information visualization to tell the desired story.

The *ntv* module serves as a tailored solution for this particular scenario. Our module is based on two main components, *NarrativeTextSpec* and *NarrativeTextVis*. NarrativeTextSpec is utilized to declare standard JSON structures for text collections. Its outputs are named as *ntv-schema* which presents the component information of insights such as context, contradiction, attribution, and recommendation precisely to the readers (See Section 4.1.4 for details). This JSON schema can be widely applied in numerous automated (including AI-generated content) scenarios and offers good extensibility. NarrativeTextVis provides a react rendering component that can be employed to visualize any text specification described using *ntv-schema*. See the text elements in the middle of Fig. 7 for an example.

Specifically, when processing textual information, it encompasses two primary processes. Firstly, the NarrativeTextSpec component standardizes the definition of textual interpretation and constructs the *ntv-schema* based on the input. Then, the *ntv-schema* is passed to the NarrativeTextVis component for the application of text visualization mapping, resulting in the display of the interpreted text. When the required content types of the text exceed the default phrase and paragraph types of *ntv-schema*, customization of plugins can be employed. By extending the display and interaction through custom plugins, nearly all scenarios can be accomplished. Furthermore, the framework offers the *InsightCard* based on *ntv*, enabling the presentation of data insights with rich visuals and explanatory charts in the form of captivating cards, as shown in Fig. 7.

4.3. Extensible capability encapsulation

The framework also provides a plug-and-play React component library based on the integration of AVA capabilities. This library consists of the *NarrativeTextVis*, which helps present data insights in text format, the *AutoChart*, which automatically suggests and renders the right chart, and the *InsightCard*, which displays data insight in a combination of graphics and text.

5. Evaluation

5.1. Case study

We conduct case studies based on the empiric- and insight-driven recommendation pipeline respectively with four analysts and designers from the industry (P1-P4). They all have over three years of project experience in visual design and data analysis.

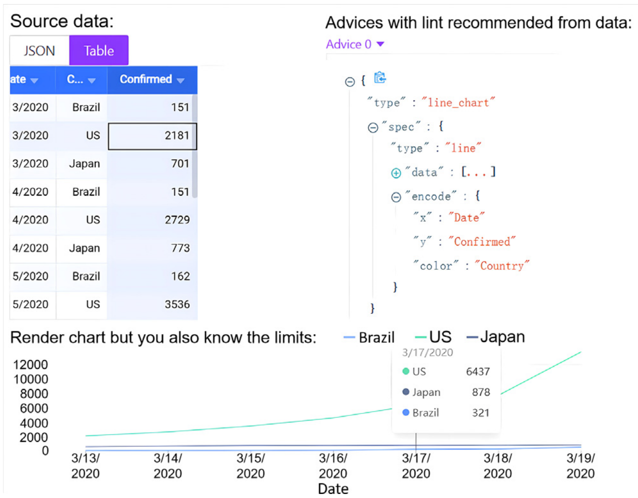

Fig. 8. An instance of empiric-driven recommendation process for user P1.

5.1.1. Empiric-driven recommendation

In terms of our empiric-driven visualization recommendation model, all users agree that our tool contributed a lot on raising the expressiveness of charts, which significantly reduces the time cost when choosing appropriate charts based on empirical rules.

Fig. 8 shows an instance of how the expert P1 employs our empiric-driven recommendation tool. In this case, P1 selected two columns of data from a dataset that encompassed the cumulative number of confirmed cases, fatalities, and recoveries attributed to Covid-19 in three countries over the course of a week. In the past P1 would have to try different kinds of charts to visualize them better. However, our tool can generate recommended line charts to illustrate the trends in the number of confirmed cases within the data, immediately when the columns are loading. In addition, users can interact with the generated charts to obtain specific data information about a particular section within the graph. All our users agree that this recommendation is helpful in their usage scenarios.

5.1.2. Insight-driven recommendation

When it comes to the insight-driven visualization recommendation tool, the main feedback is that our tool can help users understand the overview of data quickly and significantly reduce the time cost for exploratory analysis.

Fig. 7 shows the process when the expert P2 is conducting exploratory analysis within the dataset via our insight-driven recommendation tool. Under this circumstance, P2 loaded an unfamiliar dataset, which pertains to the daily discounted prices over a certain period of time. In the past P2 would spend tens of minutes to comprehend the overall nature of the data by using different analytical and visualization tools. Using our recommendation process, we can show P2 overall insights extracted from the whole dataset, including the trend and outliers. According to users' feedback, the insight-driven process can help them understand new and unknown datasets faster and easier.

5.2. Comparative analysis

We compared *AVA* to several commonly used data visualization tools, libraries, and systems in the industry across various dimensions, including Tableau (Mackinlay et al., 2007), Echarts (Li et al., 2018), Vega-Lite (Satyanarayan et al., 2016), D3.js (Bostock et al., 2011), Matplotlib (Hunter, 2007), and Draco (Moritz et al., 2018). The following presents the results of the comparison (see Table 1).

In contrast to other JavaScript visualization libraries, AVA embodies intelligence at its core. It not only simplifies chart configuration, making it more user-friendly than D3.js but also surpasses it by offering features such as insight mining and chart recommendations. These tools are implemented naturally with a few extra lines of code, introducing an auto-suggest functionality for chart types and specifics according to data and analytical demands - a trait that is often absent in conventional visualization tools.

When compared to alternate types of tools like Python libraries and software like Draco, AVA clearly comes out as a beginner-friendly option. One of its strengths is that it is very straightforward and transparent in its functions, making it easier for engineers to learn and use. Comparatively, although Tableau is user-friendly with its drag-and-drop feature, it has limitations in its visual designs and customization options. AVA allows users to adapt charts based on their data and needs, resulting in intelligently personalized visualizations.

6. Discussion

In this section, we discuss the existing impact of *AVA*, its limitations, and shed light on future work.

6.1. Impact

The official 1.0 version of *AVA* was released on November 22, 2020, and has gone through several iterations since then. *AVA* currently has over 1.3k stars on Github, reflecting its widespread popularity. Since *AVA* is open source, community contributors can also submit code to improve *AVA* or develop their own products based on it. As a result, *AVA*'s features are constantly enriched and improved. Further, *AVA* is widely adopted in business intelligence scenarios. For example, the Ant Group's internal business intelligence platform DeepInsight uses *AVA* to automate the mining of insights from large-scale data and ChartCube,[10] an online chart generation tool, also uses *AVA* to let users quickly create visualization charts by drag and drop.

6.2. Limitations

Despite its innovative features, *AVA* also has a few limitations worth noting. First, it lacks support for configuration recommendations, leaving the task of understanding and setting configurations to the user. This could complicate the user experience, particularly for those new to data visualization (Stolper et al., 2014). Second, *AVA* currently focuses on recommending statistical charts. In particular, it does not provide comprehensive solutions for complex chart types such as maps, graphs, or juxtaposed charts, limiting its potential for higher levels of visualization comprehension (Quadri et al., 2024). Another ongoing challenge is to expand *AVA*'s coverage and recommendation capabilities while maintaining its user-friendliness and ease of use with regarding open-course software best practices (Wilson et al., 2017).

6.3. Future work

AVA has made initial attempts at intelligent visualization so far, and there are many valuable directions for future development. First, we plan to summarize more design experience from visualization academia and industry for better visualization recommendation and linter. Second, the underlying declarative syntax across AntV will continue to be improved, and interaction syntax will be added to it so that interactions in visualization

10 https://chartcube.alipay.com/.

Table 1
Comparison of visualization tools.

Library	Insight mining	Chart recommendation	Usability	Customizability
AVA	✓	✓	Moderate	Moderate
ECharts	✗	✗	Moderate	Moderate
Vega-Lite	✗	✗	Moderate	Moderate
D3.js	✗	✗	Hard	Powerful
Tableau	✗	✓	Easy	Limited
Matplotlib	✗	✗	Hard	Powerful
Draco	✗	✓	Moderate	Moderate

charts can be recommended. Finally, the recommended visualization charts are currently mainly limited to regular diagrams, and we intend to add recommendations for visual forms such as maps and graphs to suit more practical application scenarios.

7. Conclusion

This paper introduces *AVA*, an open-source framework that addresses the complexities of modern data visualization. *AVA* combines empiric and insight-driven methods, offering visually appealing and meaningful charts and narrative insight descriptions. Through cases and comparative analysis, we show that *AVA* enables effective visualization recommendation with flexible customization. As an open-source tool, *AVA* invites continuous improvement for further refinement in tackling more complex data and scenarios.

CRediT authorship contribution statement

Jiazhe Wang: Software, Writing – original draft. **Xi Li:** Software. **Chenlu Li:** Software, Writing – review & editing. **Di Peng:** Software. **Arran Zeyu Wang:** Software, Writing – review & editing. **Yuhui Gu:** Software. **Xingui Lai:** Software. **Haifeng Zhang:** Software. **Xinyue Xu:** Software. **Xiaoqing Dong:** Conceptualization. **Zhifeng Lin:** Conceptualization. **Jiehui Zhou:** Writing – original draft. **Xingyu Liu:** Writing – original draft. **Wei Chen:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

We would like to thank Jiefeng Yuan, Fuling Sun, Nan Chen, Ying Chen, Peiying Zhang and Liangjing Bi for their kind help on this paper. Wei Chen is supported by National Natural Science Foundation of China (62132017) and Zhejiang Provincial Natural Science Foundation of China (LD24F020011).

Ethical approval

This study does not contain any studies with human or animal subjects performed by any of the authors.

References

Ananthanarayanan, R., Lohia, P.K., Bedathur, S., 2018. Datavizard: Recommending visual presentations for structured data. In: Proc. 21st International Workshop on the Web and Databases. pp. 1–6.
Armstrong, Z., Wattenberg, M., 2014. Visualizing statistical mix effects and simpson's paradox. IEEE Trans. Vis. Comput. Graphics 20 (12), 2132–2141.
Basole, R.C., Major, T., 2024. Generative AI for visualization: Opportunities and challenges. IEEE Comput. Graph. Appl. 44 (2), 55–64.
Battle, L., Ottley, A., 2023. What do we mean when we say "insight"? A formal synthesis of existing theory. IEEE Trans. Vis. Comput. Graphics.
Benesty, J., Chen, J., Huang, Y., Cohen, I., 2009. Pearson correlation coefficient. In: Noise Reduction in Speech Processing. Springer, pp. 1–4.
Borland, D., Wang, A.Z., Gotz, D., 2024. Using counterfactuals to improve causal inferences from visualizations. IEEE Comput. Graph. Appl. 44 (1), 95–104.
Bostock, M., Ogievetsky, V., Heer, J., 2011. D3: Data-driven documents. IEEE Trans. Vis. Comput. Graphics 17 (12), 2301–2309.
Brown, E.T., Ottley, A., Zhao, H., Lin, Q., Souvenir, R., Endert, A., Chang, R., 2014. Finding waldo: Learning about users from their interactions. IEEE Trans. Vis. Comput. Graphics 20 (12), 1663–1672.
Ceneda, D., Gschwandtner, T., May, T., Miksch, S., Schulz, H.-J., Streit, M., Tominski, C., 2016. Characterizing guidance in visual analytics. IEEE Trans. Vis. Comput. Graphics 23 (1), 111–120.
Chen, Q., Cao, S., Wang, J., Cao, N., 2023. How does automation shape the process of narrative visualization: A survey of tools. IEEE Trans. Vis. Comput. Graphics.
Chen, W., Huang, Z., Wu, F., Zhu, M., Guan, H., Maciejewski, R., 2017. VAUD: A visual analysis approach for exploring spatio-temporal urban data. IEEE Trans. Vis. Comput. Graphics 24 (9), 2636–2648.
Chen, Q., Sun, F., Xu, X., Chen, Z., Wang, J., Cao, N., 2021. Vizlinter: A linter and fixer framework for data visualization. IEEE Trans. Vis. Comput. Graphics 28 (1), 206–216.
Cui, Z., Badam, S.K., Yalçin, M.A., Elmqvist, N., 2019a. Datasite: Proactive visual data exploration with computation of insight-based recommendations. Inf. Vis. 18 (2), 251–267.
Cui, W., Zhang, X., Wang, Y., Huang, H., Chen, B., Fang, L., Zhang, H., Lou, J.-G., Zhang, D., 2019b. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. IEEE Trans. Vis. Comput. Graphics 26 (1), 906–916.
Demiralp, Ç., Haas, P.J., Parthasarathy, S., Pedapati, T., 2017. Foresight: Recommending visual insights. Proc. VLDB J..
Dibia, V., 2023. LIDA: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models. arXiv:2303.02927.
Dibia, V., Demiralp, Ç., 2019. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. IEEE Comput. Graph. Appl. 39 (5), 33–46.
Dimara, E., Bailly, G., Bezerianos, A., Franconeri, S., 2018. Mitigating the attraction effect with visualizations. IEEE Trans. Vis. Comput. Graphics 25 (1), 850–860.
Ding, R., Han, S., Xu, Y., Zhang, H., Zhang, D., 2019. Quickinsights: Quick and automatic discovery of insights from multi-dimensional data. In: Proc. 2019 ACM SIGMOD Conference. pp. 317–332.
Gleicher, M., Albers, D., Walker, R., Jusufi, I., Hansen, C.D., Roberts, J.C., 2011. Visual comparison for information visualization. Inf. Vis. 10 (4), 289–309.
Gotz, D., Stavropoulos, H., 2014. Decisionflow: Visual analytics for high-dimensional temporal event sequence data. IEEE Trans. Vis. Comput. Graphics 20 (12), 1783–1792.
Gotz, D., Sun, S., Cao, N., 2016. Adaptive contextualization: Combating bias during high-dimensional visualization and data selection. In: Proc. 21st ACM IUI Conference. pp. 85–95.
Gotz, D., Wen, Z., 2009. Behavior-driven visualization recommendation. In: Proc. 14th ACM IUI Conference. pp. 315–324.
Gotz, D., Zhou, M.X., 2009. Characterizing users' visual analytic activity for insight provenance. Inf. Vis. 8 (1), 42–55.
Guo, S., Xu, K., Zhao, R., Gotz, D., Zha, H., Cao, N., 2017. Eventthread: Visual summarization and stage analysis of event sequence data. IEEE Trans. Vis. Comput. Graphics 24 (1), 56–65.
Hu, K., Bakker, M.A., Li, S., Kraska, T., Hidalgo, C., 2019. Vizml: A machine learning approach to visualization recommendation. In: Proc. 2019 CHI Conf. Hum. Factors Comput. Syst.. pp. 1–12.
Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. Comput. Sci. Eng. 9 (3), 90–95.

Jin, Z., Guo, S., Chen, N., Weiskopf, D., Gotz, D., Cao, N., 2020. Visual causality analysis of event sequence data. IEEE Trans. Vis. Comput. Graphics 27 (2), 1343–1352.
Kale, A., Guo, Z., Qiao, X.L., Heer, J., Hullman, J., 2023. Evm: Incorporating model checking into exploratory visual analysis. IEEE Trans. Vis. Comput. Graphics.
Kaul, S., Borland, D., Cao, N., Gotz, D., 2021. Improving visualization interpretation using counterfactuals. IEEE Trans. Vis. Comput. Graphics 28 (1), 998–1008.
Li, D., Mei, H., Shen, Y., Su, S., Zhang, W., Wang, J., Zu, M., Chen, W., 2018. Echarts: A declarative framework for rapid construction of web-based visualization. Vis. Inf. 2 (2), 136–146.
Li, G., Wang, X., Aodeng, G., Zheng, S., Zhang, Y., Ou, C., Wang, S., Liu, C.H., 2024. Visualization generation with large language models: An evaluation. arXiv:2401.11255.
Lin, Q., Ke, W., Lou, J.-G., Zhang, H., Sui, K., Xu, Y., Zhou, Z., Qiao, B., Zhang, D., 2018. Bigin4: Instant, interactive insight identification for multi-dimensional big data. In: Proc. 24th ACM KDD Conference. pp. 547–555.
Lin, H., Moritz, D., Heer, J., 2020. Dziban: Balancing agency & automation in visualization design via anchored recommendations. In: Proc. 2020 CHI Conf. Hum. Factors Comput. Syst.. pp. 1–12.
Ma, P., Ding, R., Han, S., Zhang, D., 2021. MetaInsight: Automatic discovery of structured knowledge for exploratory data analysis. In: Proc. 2021 ACM SIGMOD Conference. pp. 1262–1274.
Ma, P., Ding, R., Wang, S., Han, S., Zhang, D., 2023a. Demonstration of InsightPilot: An LLM-empowered automated data exploration system. arXiv:2304.00477.
Ma, P., Ding, R., Wang, S., Han, S., Zhang, D., 2023b. XInsight: explainable data analysis through the lens of causality. Proc. ACM SIGMOD Conf. 1 (2), 1–27.
Ma, R., Mei, H., Guan, H., Huang, W., Zhang, F., Xin, C., Dai, W., Wen, X., Chen, W., 2020. Ladv: Deep learning assisted authoring of dashboard visualizations from images and sketches. IEEE Trans. Vis. Comput. Graphics 27 (9), 3717–3732.
Mackinlay, J., 1986. Automating the design of graphical presentations of relational information. ACM Trans. Graph. 5 (2), 110–141.
Mackinlay, J., Hanrahan, P., Stolte, C., 2007. Show me: Automatic presentation for visual analysis. IEEE Trans. Vis. Comput. Graphics 13 (6), 1137–1144.
Mafrur, R., Sharaf, M.A., Khan, H.A., 2018. Dive: Diversifying view recommendation for visual data exploration. In: Proc. 27th ACM CIKM Conference. pp. 1123–1132.
Moritz, D., Wang, C., Nelson, G.L., Lin, H., Smith, A.M., Howe, B., Heer, J., 2018. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. IEEE Trans. Vis. Comput. Graphics 25 (1), 438–448.
Nanayakkara, P., Kim, H., Wu, Y., Sarvghad, A., Mahyar, N., Miklau, G., Hullman, J., 2024. Measure-observe-remeasure: An interactive paradigm for differentially-private exploratory analysis. In: IEEE Symp. S&P. pp. 231–231.
North, C., 2006. Toward measuring visualization insight. IEEE Comput. Graph. Appl. 26 (3), 6–9.
Qin, X., Luo, Y., Tang, N., Li, G., 2020. Making data visualization more efficient and effective: a survey. Proc. VLDB J. 29 (1), 93–117.
Quadri, G.J., Rosen, P., 2021. A survey of perception-based visualization studies by task. IEEE Trans. Vis. Comput. Graphics.
Quadri, G.J., Wang, A.Z., Wang, Z., Adorno, J., Rosen, P., Szafir, D.A., 2024. Do you see what I see? A qualitative study eliciting high-level visualization comprehension. In: Proc. 2024 CHI Conf. Hum. Factors Comput. Syst.. pp. 1–26.
Rey, B., Lee, B., Choe, E.K., Irani, P., 2024. Databiting: Lightweight, transient, and insight rich exploration of personal data. IEEE Comput. Graph. Appl. 44 (2), 65–72.
Satyanarayan, A., Moritz, D., Wongsuphasawat, K., Heer, J., 2016. Vega-lite: A grammar of interactive graphics. IEEE Trans. Vis. Comput. Graphics 23 (1), 341–350.
Satyanarayan, A., Russell, R., Hoffswell, J., Heer, J., 2015. Reactive vega: A streaming dataflow architecture for declarative interactive visualization. IEEE Trans. Vis. Comput. Graphics 22 (1), 659–668.
Shi, D., Sun, F., Xu, X., Lan, X., Gotz, D., Cao, N., 2021. Autoclips: An automatic approach to video generation from data facts. In: Comput. Graph. Forum. Vol. 40, Wiley Online Library, pp. 495–505.
Shi, D., Xu, X., Sun, F., Shi, Y., Cao, N., 2020. Calliope: Automatic visual data story generation from a spreadsheet. IEEE Trans. Vis. Comput. Graphics 27 (2), 453–463.
Smart, S., Szafir, D.A., 2019. Measuring the separability of shape, size, and color in scatterplots. In: Proc. 2019 CHI Conf. Hum. Factors Comput. Syst.. pp. 1–14.
Smart, S., Wu, K., Szafir, D.A., 2019. Color crafting: Automating the construction of designer quality color ramps. IEEE Trans. Vis. Comput. Graphics 26 (1), 1215–1225.
Srinivasan, A., Drucker, S.M., Endert, A., Stasko, J., 2018. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. IEEE Trans. Vis. Comput. Graphics 25 (1), 672–681.
Stolper, C.D., Perer, A., Gotz, D., 2014. Progressive visual analytics: User-driven visual exploration of in-progress analytics. IEEE Trans. Vis. Comput. Graphics 20 (12), 1653–1662.
Szafir, D.A., Borgo, R., Chen, M., Edwards, D.J., Fisher, B., Padilla, L., 2023. Visualization Psychology. Springer Nature.
Szafir, D.A., Haroz, S., Gleicher, M., Franconeri, S., 2016. Four types of ensemble coding in data visualizations. J. Vis. 16 (5), 11.
Tang, B., Han, S., Yiu, M.L., Ding, R., Zhang, D., 2017. Extracting top-k insights from multi-dimensional data. In: Proc. 2017 ACM SIGMOD Conference. pp. 1509–1524.
Tian, Y., Cui, W., Deng, D., Yi, X., Yang, Y., Zhang, H., Wu, Y., 2024. Chartgpt: Leveraging llms to generate charts from abstract natural language. IEEE Trans. Vis. Comput. Graphics.
Tseng, C., Quadri, G.J., Wang, Z., Szafir, D.A., 2023. Measuring categorical perception in color-coded scatterplots. In: Proc. 2023 CHI Conf. Hum. Factors Comput. Syst..
Tseng, C., Wang, A.Z., Quadri, G.J., Albers Szafir, D., 2024. Revisiting categorical color perception in scatterplots: Sequential, diverging, and categorical palettes. In: Proc. EuroVis 2024 - Short Papers.
Vartak, M., Rahman, S., Madden, S., Parameswaran, A., Polyzotis, N., 2015. Seedb: Efficient data-driven visualization recommendations to support visual analytics. In: Proc. VLDB Endowment. Vol. 8, p. 2182.
Wang, A.Z., Borland, D., Gotz, D., 2024. An empirical study of counterfactual visualization to support visual causal inference. Inf. Vis. 14738716241229437.
Wang, Y., Sun, Z., Zhang, H., Cui, W., Xu, K., Ma, X., Zhang, D., 2019. Datashot: Automatic generation of fact sheets from tabular data. IEEE Trans. Vis. Comput. Graphics 26 (1), 895–905.
Wang, C., Thompson, J., Lee, B., 2023. Data formulator: AI-powered concept-driven visualization authoring. IEEE Trans. Vis. Comput. Graphics 1–11.
Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., Teal, T.K., 2017. Good enough practices in scientific computing. PLoS Comput. Biol. 13 (6), e1005510.
Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., Heer, J., 2015. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. IEEE Trans. Vis. Comput. Graphics 22 (1), 649–658.
Wongsuphasawat, K., Qu, Z., Moritz, D., Chang, R., Ouk, F., Anand, A., Mackinlay, J., Howe, B., Heer, J., 2017. Voyager 2: Augmenting visual analysis with partial view specifications. In: Proc. 2017 CHI Conf. Hum. Factors Comput. Syst.. pp. 2648–2659.
Wood, J., Dykes, J., Slingsby, A., Clarke, K., 2007. Interactive visual exploration of a large spatio-temporal dataset: Reflections on a geovisualization mashup. IEEE Trans. Vis. Comput. Graphics 13 (6), 1176–1183.
Wu, Z., Chen, W., Ma, Y., Xu, T., Yan, F., Lv, L., Qian, Z., Xia, J., 2023a. Explainable data transformation recommendation for automatic visualization. Front. Inf. Technol. Electron. Eng. 24 (7), 1007–1027.
Wu, J., Chung, J.J.Y., Adar, E., 2023b. Viz2viz: Prompt-driven stylized visualization generation using a diffusion model. arXiv preprint arXiv:2304.01919.
Wu, A., Tong, W., Dwyer, T., Lee, B., Isenberg, P., Qu, H., 2020. Mobilevisfixer: Tailoring web visualizations for mobile phones leveraging an explainable reinforcement learning framework. IEEE Trans. Vis. Comput. Graphics 27 (2), 464–474.
Xiao, S., Huang, S., Lin, Y., Ye, Y., Zeng, W., 2023. Let the chart spark: Embedding semantic context into chart with text-to-image generative model. IEEE Trans. Vis. Comput. Graphics.
Xiong, C., Shapiro, J., Hullman, J., Franconeri, S., 2019. Illusion of causality in visualized data. IEEE Trans. Vis. Comput. Graphics 26 (1), 853–862.
Yu, B., Silva, C.T., 2019. Flowsense: A natural language interface for visual data exploration within a dataflow system. IEEE Trans. Vis. Comput. Graphics 26 (1), 1–11.
Zeng, Z., Moh, P., Du, F., Hoffswell, J., Lee, T.Y., Malik, S., Koh, E., Battle, L., 2021. An evaluation-focused framework for visualization recommendation algorithms. IEEE Trans. Vis. Comput. Graphics 28 (1), 346–356.
Zhou, Z., Meng, L., Tang, C., Zhao, Y., Guo, Z., Hu, M., Chen, W., 2018. Visual abstraction of large scale geospatial origin-destination movement data. IEEE Trans. Vis. Comput. Graphics 25 (1), 43–53.
Zhou, Z., Wang, W., Guo, M., Wang, Y., Gotz, D., 2022a. A design space for surfacing content recommendations in visual analytic platforms. IEEE Trans. Vis. Comput. Graphics 29 (1), 84–94.
Zhou, J., Wang, X., Wang, J., Ye, H., Wang, H., Zhou, Z., Han, D., Ying, H., Wu, J., Chen, W., 2023. FraudAuditor: A visual analytics approach for collusive fraud in health insurance. IEEE Trans. Vis. Comput. Graphics.
Zhou, J., Wang, X., Wong, J.K., Wang, H., Wang, Z., Yang, X., Yan, X., Feng, H., Qu, H., Ying, H., et al., 2022b. Dpviscreator: Incorporating pattern constraints to privacy-preserving visualizations via differential privacy. IEEE Trans. Vis. Comput. Graphics 29 (1), 809–819.
Zhou, Z., Wen, X., Wang, Y., Gotz, D., 2021. Modeling and leveraging analytic focus during exploratory visual analysis. In: Proc. 2021 CHI Conf. Hum. Factors Comput. Syst.. pp. 1–15.