TaskVis: Task-oriented Visualization Recommendation

Leixian Shen[1,2], Enya Shen[1,2], Zhiwei Tai[1,2], Yiran Song[1,2] and Jianmin Wang[1,2]

[1]School of Software, Tsinghua University, China
[2]National Engineering Laboratory for Big Data Software, China

Abstract

General visualization recommendation systems typically make design decisions of the dataset automatically. However, these systems are only able to prune meaningless visualizations but fail to recommend targeted results. In this paper, we contributed TaskVis, a task-oriented visualization recommendation approach with detailed modeling of the user's analysis task. We first summarized a task base with 18 analysis tasks by a survey both in academia and industry. On this basis, we further maintained a rule base, which extends empirical wisdom with our targeted modeling of analysis tasks. Inspired by Draco, we enumerated candidate visualizations through answer set programming. After visualization generation, TaskVis supports four ranking schemes according to the complexity of charts, coverage of the user's interested columns and tasks. In two user studies, we found that TaskVis can well reflect the user's preferences and strike a great balance between automation and the user's intent.

CCS Concepts

• ***Human-centered computing*** → ***Information visualization;*** • ***Information systems*** → *Data analytics;*

1. Introduction

Visualization recommendation (VisRec) has attracted much attention both in academia and industry] [QLTL20, ZSJ*20]. Early works represented by APT [Mac86] leverage predefined rules to prune large search space. The rules in the community increase iteratively as subsequent works embody the previous works and produce new rules. Voyager [WMA*16], Show me [MHS07], and Profiler [KPP*12] all made contributions. On top of these works, Moritz et al. [MWN*19] proposed a formal framework Draco that models visualization design knowledge as a collection of Answer Set Programming (ASP) constraints. Recently, learning-based systems like DeepEye [LQTL18], Data2Vis [DD19], Draco-Learn [MWN*19] and VizML [HBL*19] can learn from existing well-designed examples to train models for VisRec.

Previous studies have documented that the effectiveness of a visualization largely depends on the user's analysis tasks [SED19, RAW*16, BM13, BBK*18]. However, most of the existing systems lack detailed modeling of analysis tasks, so they are only able to prune meaningless visualizations but fail to recommend targeted results. Even if some works have taken analysis tasks into account, the comprehensiveness and depth are still insufficient. For example, Moritz et al. [MWN*19] only grouped tasks into two categories: value tasks and summary tasks. NL4DV [NSS21] only integrates five types of low-level analytic tasks into VisRec system. In response to the aforementioned problem, we proposed a task-oriented VisRec approach, TaskVis. In details, our contributions are: we summarized a task base of 18 classical low-level analysis tasks with their appropriate chart types by a survey both in academia and industry. On this basis, we further maintained a rule base, which extends empirical wisdom with our targeted modeling of analysis tasks. We also realized a task-oriented recommendation engine that supports four ranking schemes. And two human-subjects experiments were conducted to evaluate the effectiveness of TaskVis.

2. TaskVis Overview

The outline of TaskVis is shown in Figure 1. Users can enter optional interested columns and analysis tasks chosen from task base at will. No input is also allowed. The preprocess module later extracts data features (e.g., data type and cardinality) automatically, and user intervention is allowed here as the data type may be ambiguous in some scenarios. After preprocessing, the extracted data features together with the user's inputs are sent to the visualization generation module. Conversion to ASP constraints is a premise to make input available for Clingo, which is an ASP system to solve logic programs. Apart from the user's input, a rule base is the core of TaskVis and also should be sent into Clingo. All enumerated candidate visualizations in ASP constraints are transformed into Vega-Lite [SMWH17] specification at the end of visualization generation stage. In the visualization ranking phase, if the visualization generation module has processed multi-tasks, the Vega-Lite list output should be de-duplicated first. After that, each Vega-Lite is assigned a cost value based on a cost map inspired by GraphScape [KWHH17]. On the basis, four rank schemes are allowed for users to select and each scheme has applicable scenarios. Finally, a VisRec list is presented to users, where each chart in the list is an independent recommendation result.

DOI: 10.2312/evs.20211061

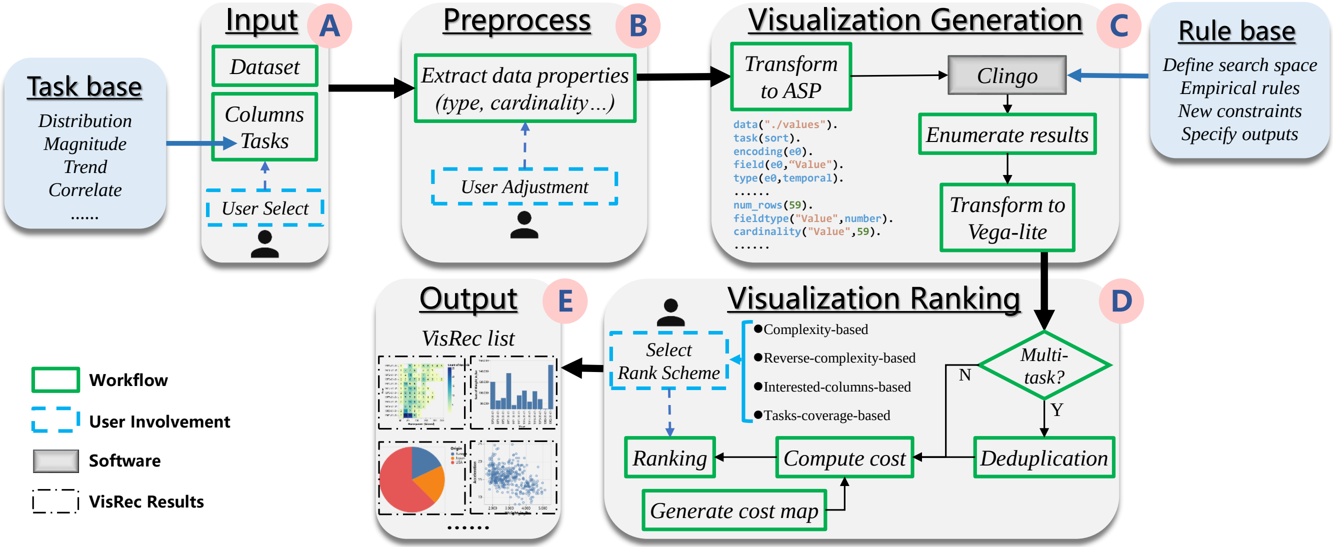

Figure 1: *Overview of TaskVis. TaskVis consists of five modules and two bases (for tasks and rules respectively). (A) Input: accepts users' input. (B) Preprocess: extract data properties. (C) Visualization Generation: enumerate all qualified candidates. (D) Visualization Ranking: rank all visualizations according to selected scheme. (E) Output: present recommendation results to users.*

3. Task base

We maintain a task base extracted from three parts: **(a) Empirical academic studies:** A large body of early works paid attention to the effectiveness of visualization types for a selected task, either special for a particular chart type or comprehensive evaluation. **(b) Guides summarized on visualization practices:** Various industries such as newspaper, finance, geography, and engineering have a strong requirement for visualization, rich practical experience has also been accumulated in the development of the industry. **(c) Customized tasks:** Apart from common tasks that have been widely analyzed, the task base also contains some others that we consider meaningful, such as *error range*. Referring to the methodology in [SG18], we summarized 18 classical analysis tasks and their suitable chart types as shown in Table 1. It should be noted that we only list representative references in the table as they have embodied most of other works.

4. Visualization Generation

VisRec systems need to enumerate all possible visualizations first and then make recommendations [QLTL20]. The entire search space for the VisRec problem is huge. Fortunately, there are many rules either from users or traditional wisdom to prune meaningless visualizations. However, if all these rules are implemented in code with branch structure, it would be very complicated to manage. Inspired by Draco [MWN*19], we leverage Answer Set Programming (ASP) to construct design rules in a unified and extensible form. ASP is a declarative programming paradigm based on logic programs and their answer sets. It provides a simple yet powerful modeling language to solve combinatorial problems. TaskVis integrates Clingo [Cli] software to solve the logic programs. We also extend empirical wisdom with our targeted modeling of analysis tasks to maintain a rule base, which consists of three parts: **(1) Empirical wisdom:** Thanks to Draco [MWN*19], which produced the Draco knowledge base as answer set programs. It embodies prior researches such as APT [Mac86]. The knowledge base includes the declaration of domains to visualization attributes, definition of search space, various constraints to restrict the search space, and output specification. **(2) Customized rules for task:** After summarizing a task base in Section 3, we first formulate rules about the marks suitable for each task, and then generate separate rules based on the characteristics of each task. **(3) User partial specification:** TaskVis allows users to input partial specifications which includes interested columns and tasks. It helps to restrict the search space within well-formed specifications of users' interests. Recall Figure 1, after transforming to ASP programs, users' input along with rule base are sent into Clingo. Clingo outputs all meaningful candidate visualizations.

5. Visualization Ranking

Inspired by GraphScape [KWHH17] which leverages linear programs to derive transition costs via a partial ordering of visualization components, we also generate a cost map to measure the complexity of different components. In details, we extend GraphScape with the supplement and subdivision of visualization components. For a chart, we define *cost score* as the sum of corresponding cost values of different components. The lower *cost score* means the visualization is less complex. After obtaining the *cost score* of each chart, four rank schemes are designed as follows:

- **(R1) Complexity-based ranking:** This rank scheme sorts visualizations by the chart's *cost score* from low to high. According to our scoring methodology, the lower the *cost score*, the easier to understand the chart, which is in line with people's habit of observing charts. This scheme is suitable for most analysis tasks, such as *characterize distribution*, *cluster*, *correlate* etc.

Table 1: *Task base. Mark column lists suitable marks, where the rank has priority, (*) indicates the combination of marks, e.g. rect(text) means a text layer is superimposed on rect chart.*

Task	Mark	Description	Reference
Change Over Time	line/area	Analyse how the data changes over time series	[Rib, CHA, SS, MR, SR]
Characterize Distribution	bar/point	Characterize the distribution of the data over the set	[AES05, SED19, Rib, CHA, SS, Abe, Ana, NSS21, SG18, WSZ*20, SDES19]
Cluster	bar/point	Find clusters of similar attribute values	[AES05, SED19, CHA, SSX*19, SDES19]
Comparison	line/point/bar	Give emphasis to comparison on different entities	[Rib, MR, Abe, Ana, SR, SG18, KH18, WSZ*20, HGH*19, HOH18]
Compute Derived Value	rect(text)/arc/bar	Compute aggregated or binned numeric derived value	[AES05, SED19, WSZ*20, HOH18, SDES19]
Correlate	bar/line	Determine useful relationships between the columns	[AES05, SED19, Rib, CHA, SS, Abe, Ana, SR, NSS21, SG18, SSX*19, WSZ*20, HOH18, SDES19]
Determine Range	tick/boxplot	Find the span of values within the set	[AES05, SED19, Rib, SDES19]
Deviation	bar(rule)/point(rule)	Compare data with certain value like zero or mean	[SS]
Error Range	errorband/errorbar	Summarizes an error range of quantitative values	[CG14]
Filter	rect/bar/arc	Find data cases satisfying the given constrains	[AES05, SED19, NSS21, WSZ*20, SDES19]
Find Anomalies	bar/point	Identify any anomalies within the dataset	[AES05, SED19, SG18, SSX*19, WSZ*20, HGH*19, SDES19]
Find Extremum	bar/point	Find extreme values of data attribute	[AES05, SED19, KH18, WSZ*20, HGH*19, SDES19]
Magnitude	arc/bar	Show relative or absolute size comparisons	[Rib, SS]
Part to Whole	arc	Show component elements of a single entity	[Rib, SS, Abe, Ana, SR, WSZ*20]
Retrieve Value	rect(text)	Find values of columns	[AES05, SED19, CHA, MR, NSS21, KH18, WSZ*20, HGH*19, SDES19]
Sort	bar	Rank data according to some ordinal metric	[AES05, SED19, SS, WSZ*20, SDES19]
Spatial	geoshape/circle(text)	Show spatial data like latitude and longitude	[Rib, CHA, SS, MR, SR]
Trend	point	Use regression or loess to show the variation trend	[Ana, WSZ*20, HOH18]

- **(R2) Reverse-complexity-based ranking:** We first adopt the unsupervised algorithm DBSCAN to cluster all the results into several categories, where visualizations in one category are similar. Then the visualizations are ranked by group according to the mean *cost score* in each category from high to low while maintaining the partial rankings within the category. The scheme is appropriate for tasks like *spatial*, *sort*, and *determine range* as these charts are relatively simple, charts with high *cost score* can present more information while humans can easily perceive.
- **(R3) Interested-columns-based ranking:** We believe that users expect a chart to display as many columns of interest as possible. So if a chart fails to show all the user's interested columns, we will assign a penalty. The new *cost score* will be: *cost score* $\div (N1/N2)$, where N1 is the number of columns included in a chart, and N2 is the number of users' interested columns. Visualizations are later sorted by R1 scheme. This scheme fits some comprehensive tasks like *magnitude* and *find anomalies*.
- **(R4) Tasks-coverage-based ranking:** For multiple tasks, we believe that if a chart is an overlap of multiple tasks' VisRec list, it must have universal meaning and should be ranked in the forefront. We define *task coverage number* for each chart as the number of tasks that are applicable to this chart. All charts are first ranked by *task coverage number*, and those with the same *task coverage number* are later partially ranked by the *cost score*.

Back to Figure 1, in (D), users are allowed to select the four rank schemes freely. (E) is a VisRec list. We also recommend the rank scheme for each task as default based on the aforementioned material in Section 3.

6. Use Cases

We now illustrate the recommendation capabilities of TaskVis in specific scenarios.

Cars: Now suppose that we are exploring a dataset of cars [CAR] with different types of columns. Various analysis tasks can be explored in this dataset. Illustrate with examples, Figure 2 lists representative results ranked in the forefront for each task. The range of displacement with different cylinders can be clearly observed with boxplot chart in *determine range* task. The pie chart in *part to whole* task reveals how the dataset can be divided into three components (Europe, Japan, and USA). Due to layout restrictions, we just show some representative charts here, complete results can be found in the supplemental material, and similarly hereinafter.

COVID-19: Imagine that we are exploring the COVID-19 dataset of U.S. [DDG20], which contains the number of daily confirmed and dead cases from Jan. to Sept. 2020. For this dataset, we may be interested in the geographical distribution of deaths. The representative recommendations in *spatial* task are shown in Fig-

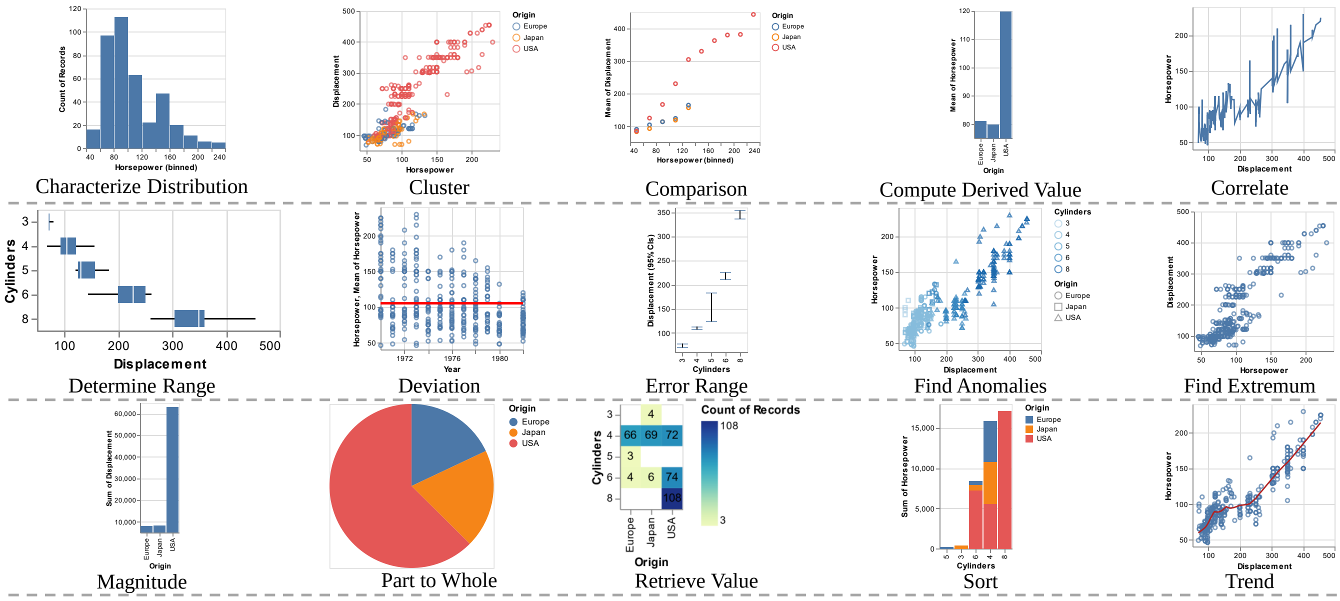

Figure 2: *Recommendations with cars dataset. Here lists representative results ranked in the forefront for each task with default rank scheme.*

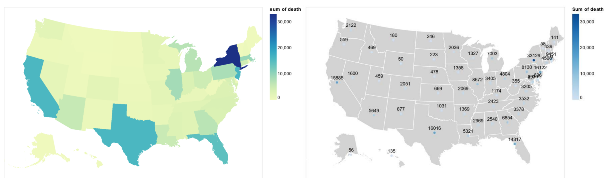

Figure 3: *Representative VisRec examples of COVID-19 dataset in spatial task.*

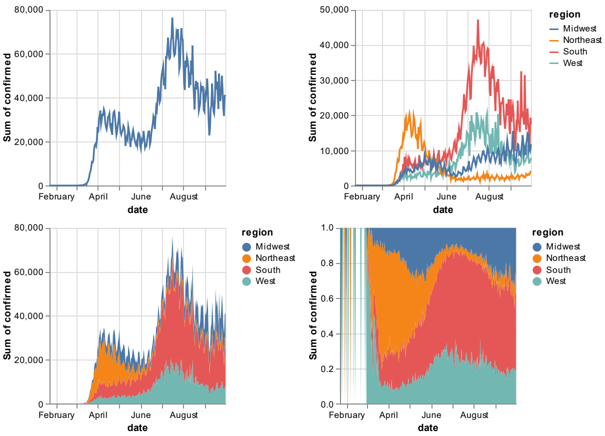

Figure 4: *Representative VisRec examples of COVID-19 dataset in change over time task.*

ure 3. Geoshape chart on the left leverages heatmap to reveal the distribution of deaths in each state, whereas the right applies color channel on circle mark, and the concrete numbers are labeled. Besides, we may also be keened on "how the number of daily confirmed changed since the epidemic?". With *change over time* task and interested columns selected, the recommendations are shown in Figure 4. We can intuitively observe the change in the number of daily confirmed (national and four regions distributed) by line and area (support stack) charts.

7. Evaluation

We conducted two user studies to evaluate TaskVis's ability of recommending targeted charts for tasks. We received effective feedbacks from 22 participants. In the first experiment, we finally collected 6249 valid cases where only 5.17% cases indicated that participants thought TaskVis's recommendations cannot reflect the analysis task at all. In the second experiment, we found that TaskVis could well reflect users' preferences. In general, TaskVis is capable of recommending targeted charts that related to users' tasks, striking a great balance between automation and the user's intent. More details of the user study can be found in the supplemental material.

8. Conclusion and Future Work

In this paper, we present TaskVis, a task-oriented visualization recommendation approach. TaskVis improves the existing technologies by correlating recommendations with the user's analysis tasks. Yet, it is still in the early stage of VisRec. In the future, we will make deeper modeling of tasks. On this basis, extracting richer data properties, integrating semantics information, and designing natural language interface will be considered.

9. Acknowledgement

The authors thank the anonymous reviewers for their valuable comments. The authors would also like to thank Halden Lin and Xinhe Wang for their helpful advice.

References

[Abe] ABELA A.: Choosing a good chart. URL: https://extremepresentation.typepad.com/blog/. 3

[AES05] AMAR R., EAGAN J., STASKO J.: Low-level components of analytic activity in information visualization. In *Proceedings of the 11th IEEE Symposium on Information Visualization, INFOVIS'05.* (Minneapolis, MN, USA, 2005), IEEE, pp. 111–117. 3

[Ana] ANALYTICS J.: Chart Chooser. URL: http://labs.juiceanalytics.com/chartchooser/. 3

[BBK*18] BEHRISCH M., BLUMENSCHEIN M., KIM N. W., SHAO L., EL-ASSADY M., FUCHS J., SEEBACHER D., DIEHL A., BRANDES U., PFISTER H., SCHRECK T., WEISKOPF D., KEIM D. A.: Quality Metrics for Information Visualization. *Computer Graphics Forum 37*, 3 (jun 2018), 625–662. 1

[BM13] BREHMER M., MUNZNER T.: A Multi-Level Typology of Abstract Visualization Tasks. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (dec 2013), 2376–2385. 1

[CAR] Cars dataset. URL: http://stat-computing.org/dataexpo/1983.html. 3

[CG14] CORRELL M., GLEICHER M.: Error Bars Considered Harmful: Exploring Alternate Encodings for Mean and Error. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2142–2151. 3

[CHA] ChartGuide poster. URL: https://chart.guide/topics/chartguide-poster-4-0/. 3

[Cli] CLINGO: Answer Set Programming. URL: https://potassco.org/. 2

[DD19] DIBIA V., DEMIRALP C.: Data2Vis: Automatic Generation of Data Visualizations Using Sequence-to-Sequence Recurrent Neural Networks. *IEEE Computer Graphics and Applications 39*, 5 (sep 2019), 33–46. 1

[DDG20] DONG E., DU H., GARDNER L.: An interactive web-based dashboard to track COVID-19 in real time. *The Lancet Infectious Diseases 20*, 5 (may 2020), 533–534. 3

[HBL*19] HU K., BAKKER M. A., LI S., KRASKA T., HIDALGO C.: VizML: A machine learning approach to visualization recommendation. In *Proceedings of the 37rd Annual ACM Conference on Human Factors in Computing Systems, CHI'19* (Glasgow, Scotland, UK, 2019), ACM Press, pp. 1–12. 1

[HGH*19] HU K., GAIKWAD S. S., HULSEBOS M., BAKKER M. A., ZGRAGGEN E., HIDALGO C., KRASKA T., LI G., SATYANARAYAN A., DEMIRALP Ç.: VizNet: Towards a large-scale visualization learning and benchmarking repository. In *Proceedings of the 37rd Annual Conference on Human Factors in Computing Systems, CHI'19* (Glasgow, UK, 2019), ACM, pp. 1–12. 3

[HOH18] HU K., ORGHIAN D., HIDALGO C.: DIVE: A mixed-initiative system supporting integrated data exploration workflows. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics, HILDA'2018* (Houston, TX, USA, 2018), ACM Press, pp. 1–7. 3

[KH18] KIM Y., HEER J.: Assessing Effects of Task and Data Distribution on the Effectiveness of Visual Encodings. *Computer Graphics Forum 37*, 3 (jun 2018), 157–167. 3

[KPP*12] KANDEL S., PARIKH R., PAEPCKE A., HELLERSTEIN J. M., HEER J.: Profiler: Integrated statistical analysis and visualization for data quality assessment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI'12* (Capri Island, Italy, 2012), ACM Press, pp. 547–554. 1

[KWHH17] KIM Y., WONGSUPHASAWAT K., HULLMAN J., HEER J.: GraphScape: A model for automated reasoning about visualization similarity and sequencing. In *Proceedings of the 35rd Annual ACM Conference on Human Factors in Computing Systems, CHI'17* (Denver, CO, USA, may 2017), ACM, pp. 2628–2638. 1, 2

[LQTL18] LUO Y., QIN X., TANG N., LI G.: Deepeye: towards automatic data visualization. In *Proceedings of the 34th IEEE International Conference on Data Engineering, ICDE'18* (Paris, France, apr 2018), IEEE, pp. 101–112. 1

[Mac86] MACKINLAY J.: Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics 5*, 2 (apr 1986), 110–141. 1, 2

[MHS07] MACKINLAY J., HANRAHAN P., STOLTE C.: Show Me: Automatic Presentation for Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (nov 2007), 1137–1144. 1

[MR] MARK L., REEVES C.: Visual Analytics. URL: https://www.pinterest.com/pin/20125529565819990/. 3

[MWN*19] MORITZ D., WANG C., NELSON G. L., LIN H., SMITH A. M., HOWE B., HEER J.: Formalizing Visualization Design Knowledge as Constraints: Actionable and Extensible Models in Draco. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (jan 2019), 438–448. 1, 2

[NSS21] NARECHANIA A., SRINIVASAN A., STASKO J.: NL4DV: A Toolkit for Generating Analytic Specifications for Data Visualization from Natural Language Queries. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (feb 2021), 369–379. 1, 3

[QLTL20] QIN X., LUO Y., TANG N., LI G.: Making data visualization more efficient and effective: a survey. *The VLDB Journal 29*, 1 (2020), 93–117. 1, 2

[RAW*16] RIND A., AIGNER W., WAGNER M., MIKSCH S., LAMMARSCH T.: Task Cube: A three-dimensional conceptual space of user tasks in visualization design and evaluation. *Information Visualization 15*, 4 (oct 2016), 288–300. 1

[Rib] RIBECCA S.: The Data Visualization Catalogue. URL: https://datavizcatalogue.com/ZH/{%}0A. 3

[SDES19] SRINIVASAN A., DRUCKER S. M., ENDERT A., STASKO J.: Augmenting Visualizations with Interactive Data Facts to Facilitate Interpretation and Communication. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (jan 2019), 672–681. 3

[SED19] SAKET B., ENDERT A., DEMIRALP C.: Task-Based Effectiveness of Basic Visualizations. *IEEE Transactions on Visualization and Computer Graphics 25*, 7 (jul 2019), 2505–2512. 1, 3

[SG18] SARIKAYA A., GLEICHER M.: Scatterplots: Tasks, Data, and Designs. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (jan 2018), 402–412. 2, 3

[SMWH17] SATYANARAYAN A., MORITZ D., WONGSUPHASAWAT K., HEER J.: Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (jan 2017), 341–350. 1

[SR] SCHWABISH J., RIBECCA S.: The Graphic Continuum. URL: https://policyviz.com/2014/09/09/graphic-continuum/. 3

[SS] STABE M., SMITH A.: Visual Vocabulary. URL: https://github.com/ft-interactive/chart-doctor/tree/master/visual-vocabulary. 3

[SSX*19] SHI D., SHI Y., XU X., CHEN N., FU S., WU H., CAO N.: Task-Oriented Optimal Sequencing of Visualization Charts. In *IEEE Visualization in Data Science, VDS'19* (oct 2019), IEEE, pp. 58–66. 3

[WMA*16] WONGSUPHASAWAT K., MORITZ D., ANAND A., MACKINLAY J., HOWE B., HEER J.: Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (jan 2016), 649–658. 1

[WSZ*20] WANG Y., SUN Z., ZHANG H., CUI W., XU K., MA X., ZHANG D.: DataShot: Automatic Generation of Fact Sheets from Tabular Data. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (jan 2020), 895–905. 3

[ZSJ*20] ZHU S., SUN G., JIANG Q., ZHA M., LIANG R.: A survey on automatic infographics and visualization recommendations. *Visual Informatics 4*, 3 (2020), 24–40. 1