More Like Vis, Less Like Vis: Comparing Interactions for Integrating User Preferences into Partial Specification Recommenders

Grace Guo, Subhajit Das, Jian Zhao, and Alex Endert

Abstract—Visualization recommendation systems make data exploration less tedious by automating the process of visualization generation. They are particularly helpful for non-expert users who may not be familiar with a data set or the process of visualization specification. These systems allow users to input their preferences in the form of partial specifications to steer the recommendations made. However, the interaction approaches for partial specification input and their trade-offs have not been explored in prior work. In this paper, we compare three different combinations of interaction approaches and granularities for users to indicate a preferred partial specification: 1) manual input, 2) inferring preferred partial specifications from binary like/dislike ratings for a visualization as a whole, or 3) inferring preferred partial specifications from binary like/dislike ratings for granular components of a visualization specification. In a between-subjects study, participants were assigned to one of three conditions and asked to complete a data exploration task. Our results indicate that manual input led to a greater coverage of data dimensions, while like/dislike ratings led to a greater diversity of marks and channels used. Qualitative participant feedback also reveals differences in user strategy and visualization comprehension across the three interaction conditions. Finally, we conclude with a discussion on implications for multiplicity and visualization comprehension during visual data exploration.

Index Terms—Information visualization, Visualization systems and software, Machine learning

1 Introduction

VISUAL data exploration is an early-stage analysis task where analysts explore new data sets, update mental models, and form new hypotheses for subsequent testing [1], [2], [3], [4], [5], [6]. To create visualizations for data exploration, analysts must decide which chart types, variables, and encodings to use, a process that involves choosing from a large number of possible options. Visualization recommendation systems (e.g., [7], [8]) were thus developed to support data exploration by suggesting visualizations that analysts may be interested in. Users can steer recommendations by inputting their preferences as a set of constraints, also called *partial specifications* [9], [10], [11], [12]. The system solves for these constraints, thus generating recommendations that satisfy user preferences. This process helps to account for individual differences [13] and ensures that recommendations are meaningful and relevant.

Partial specifications have been widely adopted in visualization recommendation systems. Tools such as CompassQL [9], Draco [10], [14], and Dziban [15] allow users to input partial specifications in the syntax of their respective query languages. Other systems such as Voyager [11] and Tableau's Show Me [16] have developed graphical user interfaces where users manually click and drag menu options to create partial specifications. Collectively, these tools adopt a manual input interaction. Prior visualization surveys have also referred to them as "knowledge-based automated visualization design techniques, which use a series of user-defined constraints and visualization design constraints to guide aesthetic and expressive visualization recommendations" [17], [18].

Manual inputs are not the only method by which users can steer machine learning models according to their preferences. In some training and fine-tuning approaches, for example, users may be asked to rate or rank multiple outputs (e.g. [19], [20], [21], [22]), or judge a single output based on an explicit set of criteria provided by researchers (e.g. [19], [23]). More recently, binary like/dislike ratings have also gained popularity as they are "more abundant, cheaper, and faster to collect" [24] and "produced by users at a much larger scale" [25]. In visualization research, binary interaction methods have been successfully adopted in recommendation systems. VizDeck [26] and VizAssist [7], for example, are two systems that use like/dislike interactions to recommend visualizations for exploration. However, unlike our work, VizDeck and VizAssist focus on learning user preferences for data statistics and analysis tasks over preferences for visual characteristics.

Taken together, these prior systems demonstrate the different ways interactive human feedback can be used to align model outputs to human expectations. However, few visualization studies have explored the tradeoffs between different interaction methods for inputting or adjusting partial specifications in recommendation systems. Manual input, as commonly seen in knowledge-based recommenders, grants analysts a high degree of control over the visualizations recommended, particularly when focusing on specific questions [11]. To effectively work with manual input recommenders, users must have sufficient domain knowledge

- *Grace Guo is with Harvard University, Cambridge, MA 02138*
- *Subhajit Das is with Amazon.*
- *Jian Zhao is with the University of Waterloo.*
- *Alex Endert is with Georgia Institute of Technology.*

Manuscript received Feb 8, 2022; revised .

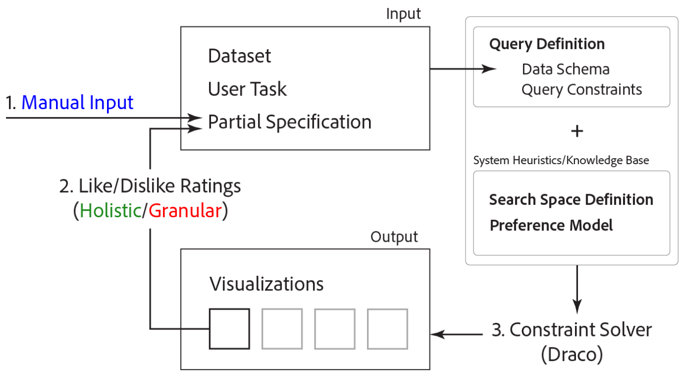

Fig. 1: An overview of partial specification-style visualization recommenders, adapted from [10]. In this paper, we compare different interaction methods (manual input/holistic binary feedback/granular binary feedback) by which users can input preferred partial specifications into the system.

to specify their preferences as a set of constraints [27]. In contrast, binary like/dislike interactions do not have a similar requirement. This makes for easier feedback collection [24], [25], but also comes at a trade-off since such systems suffer from the "cold-start" problem (i.e. insufficient initial likes/dislikes leading to inaccurate recommendations at the start) and specialization (i.e. recommendations are too similar and new items are not discovered) [27], [28].

In this paper, we present a comparative study of interaction methods for integrating user preferences into visualization recommendation systems. We use Draco [10], [14] for recommendation generation. Draco accepts user preferences as partial specifications, which are then solved to create complete specifications of Vega-lite visualizations [29]. We explore three interaction methods for partial specification: manual input, inferring preferred partial specifications from holistic binary judgments, and inferring preferred partial specifications from granular binary judgments (see Figure 1). The manual input recommender system asks users to create partial specifications using drag-and-drop interactions. In the holistic system, users provide binary like/dislike judgments for each visualization as a whole, while the granular system asks users to like/dislike parts of each visualization separately. In both the holistic and granular conditions, like/dislike ratings are used as labels to infer users' preferred partial specifications. Over many rounds of feedback, the system learns the partial specification(s) that best align with user preferences. These are then solved to generate more relevant recommendations.

We conducted a user study to compare the three interaction conditions. The study focused on non-expert users, who may be less familiar with the process of visualization specification and may thus benefit more from automated recommendations. Participants completed a data exploration task using one of the recommenders. Our hypotheses are:

H1: The manual input recommender would lead to less specialization, helping users explore a wider diversity of visualizations. Participants use a greater number of *data dimensions*, *mark types*, and *encoding channels* on average than participants who use the holistic and granular recommenders.

H2: The granularity of binary like/dislike feedback will not affect exploration outcomes. Participants who use the holistic and granular recommenders will explore a similar number of *data dimensions*, *mark types*, and *encoding channels* on average.

A quantitative analysis of study results found that manual input led to increased data dimension coverage, but like/dislike interactions led to greater encoding diversity (mark types and channels), thus rejecting H1. We also found that there was overlap between the holistic and granular recommenders in terms of average number of *data dimensions*, *mark types*, and *encoding channels* explored, which is insufficient to reject H2. However, qualitative feedback revealed differences in user strategy, visualization comprehension, and system learnability across all three interaction conditions. We also found that lack of familiarity was a key contributor to the smaller diversity of mark types and encoding channels explored in the manual input recommender.

In sum, our contributions are: 1) a comparative user study of interaction methods for partial specification in visualization recommendation systems, 2) a characterization of how interaction methods affect data exploration and the associated trade-offs, and 3) a discussion of the implications for exploration multiplicity and visualization comprehension during visual data exploration.

2 Related Work

Visualization recommendation systems formalize visualization guidelines and graphical perception knowledge as rules/constraints to help users design effective visualizations that are built on established best practices [9], [10], [14], graphical perception studies [12], and domain knowledge [30]. Of these, some systems are purely data-driven or rule-driven, recommending visualizations based on characteristics of the data set with little to no user input [31], [32], [33], [34], [35], [36], [37], [38], [39], [40] (see [17] for an overview). However, most visualization recommenders are interactive systems that also learn constraints based on user preferences. Users interact with recommendation systems using one of two mechanisms – explicit or implicit feedback. Explicit feedback is input provided by the user for the purpose of indicating their preferences, while implicit feedback infers user preferences from other user behavior and interactions [41], [42], [43]. Although a few visualization recommendation systems have been designed to make use of implicit feedback ([44], [45], [46]), the majority require some form of explicit user feedback to constrain the recommended visualizations.

2.1 Manual Input Recommenders

The most common types of explicit feedback mechanisms are manual input interactions. In visualization tools, they have also been referred to as knowledge-based techniques [17], which require explicitly defined constraints. Constraints are rules that a system tries to satisfy that are derived from visualization best practices or user preferences [7], [10], [11], [16]. Users can interact with a menu or control panel to select their preferred encodings, data, or tasks, which are then interpreted by the system as constraints for recommendation generation. For example, Tableau's Show Me feature allows users to select data dimension(s) of interest [16], while DIVE [47] and TaskVis [48] help users explore visualizations based on their tasks. The system then generates recommendations by enumerating through possible visualization specifications [10], [49] and ranking them based on how well various constraints are satisfied. By interactively specifying and updating well-defined constraints, users can steer visualization recommendation systems based on their preferences.

Also of note are visualization recommendation systems that explore interaction modalities beyond code-based or graphical user interfaces. NL4DV [50] and DracoGPT [51], for example, allow users to query a data set using natural language. For an overview of natural language interfaces, please see [52]. In addition to text, images have also been accepted as user input. In work by Chen et al. [37], users uploaded bitmap visualizations similar to their preferences to steer recommendations.

Our work draws inspiration from partial specification recommenders, such as Voyager [8], [11], Draco [10], [14], Dziban [15], and CompassQL [9]. In these systems, users manually input their preferences as partial specifications, which are interpreted by the systems as constraints to generate new recommendations.

2.2 Binary Feedback Recommenders

Compared to the prominence of manual input recommenders, there have been relatively few visualization recommenders that make use of binary feedback mechanisms. Binary feedback recommenders learn user preferences based on interactive feedback such as like/dislike ratings.

VizAssist [7] is one example of a visualization recommendation system that utilizes like/dislike feedback. The tool uses an interactive genetic algorithm and like/dislike feedback to generate recommended visualizations. However, like manual input recommenders, it requires users to explicitly define their tasks and data analysis goals, which assumes a high level of expertise from users.

VizDeck [26] is another system where users can interactively provide feedback by clicking on a recommended visualization to 'promote' or 'discard'. The system learns user preferences and sorts the recommendations to prioritize visualizations that users may be more interested in. However, unlike our work, VizDeck focuses primarily on learning user preferences for data statistics such as the number of distinct values of an attribute, its coefficient of variation, and others. In contrast, preferences for visual characteristics (e.g., encoding channels used, and *xy*-axis mappings) are de-emphasized. Furthermore, VizDeck doesn't support partial specification queries or generate new visualizations – its suggestions are limited to a predefined set.

Our work adapts the binary feedback mechanism explored in these studies to partial specification recommenders. Partial specifications are user preferences expressed as constraints that are 'solved' by the system to generate recommended visualizations [10], [14]. Using binary feedback mechanisms, we compare how partial specifications of user preferences can be inferred from like/dislike ratings instead of manual input interactions.

2.3 Visualization Embedding Spaces

The problem of visualization recommendation has previously been described as a problem of navigating a visualization design space [49], [53]. This design space has been used implicitly in a number of ML-based visualization recommendation systems, such as Data2Vis [36], VisGNN [54], and VizML [33], that convert visualizations into a feature-based or vector-based representation that is then used to make new recommendations. Other systems, such as Chartseer [55], have been developed that explicitly support the interactive exploration of a visualization embedding space. Users request recommendations by clicking on a position in the visualized embedding space. This allows them to identify previously unexplored parts of the embedding space. Recent work by Zeng et al. [49] has gone further to propose a framework that characterizes recommendation algorithms based on their coverage of the visualization design space.

Our work builds on these prior studies by combining an underlying embedding space with binary feedback provided by users. Since it is impractical for users to interactively rate all possible visualizations, we use an embedding space to identify partial specifications that are nearest neighbors of specifications that were previously "liked" by users (subsection 3.3 and subsection 3.4). Furthermore, while Chartseer remained data agnostic (data dimensions are not included in the embedding space), the systems used in our study learn user-preferred data dimensions in addition to visual characteristics.

3 Interactive Partial Specification

Partial specification-style visualization recommenders combine user preferences with visualization best practices to generate recommendations that support user analyses while also incorporating visualization design knowledge (Figure 1, adapted from [10]). Visualization best practices are represented internally as system constraints. When users input their preferences as a partial specification, this input is interpreted as additional query constraints. Combining query constraints and internal constraints, the system solves for visualizations that satisfy all constraints. These solutions are then output as recommendations.

This paper focuses on query constraints. We are interested in how users interactively input their preferences as partial specification queries, which are interpreted by the system as query constraints. Other papers have studied the internal system constraints built into Draco (also called hard and soft constraints). For example, Dziban [15] incrementally improves visualization recommendations by anchoring soft constraints from previous rounds of recommendation. Zeng et al. [12] have gone further to integrate findings from graphical perception studies into Draco by updating soft constraint weights. Readers interested in internal system constraints may refer to these prior studies.

3.1 Interaction Technique and Interaction Granularity

In this paper, we focus on three different ways by which users might interactively input their preferences as partial specifications. To evaluate the effectiveness of the different interaction conditions, we built three visualization recommendation systems—a manual input system, a holistic binary

specification	values
mark types	bar, line, point, rect, tick, area
channels	x, y, color, size, shape
aggregates	bin, min, max, mean, median, sum
dimensions	quantitative, categorical, and all dimensions in dataset

TABLE 1: Mark types, channels, aggregates, and data dimension values that can make up a partial specification in the manual input, holistic, and granular recommenders.

feedback system, and a granular binary feedback system—to use in a comparative study (Fig. Figure 1). The manual input system uses the manual input interaction technique, similar to visualization tools such as Voyager [8], [11] and Tableau's Show Me [16]. Users click and drag variables in a control panel to create partial specifications.

We compare this manual input system to two binary feedback recommenders that implement like/dislike interactions at different granularities. The holistic recommender asks users to rate prior recommendations by interactively selecting like or dislike. However, there exists a possible confounding factor of interaction granularity between the manual input and holistic conditions – the manual input condition asks users to manipulate each encoding of the partial specification separately, but the holistic condition hides these encodings from the user. As such, to better distinguish the effect of interaction granularity from interaction technique, we implemented a second granular recommender that asks users to provide feedback by rating like or dislike on each encoding separately. For both the holistic and granular recommenders, the system uses the like/dislike rating to incrementally infer the partial specifications users are likely to be interested in. Over many rounds of feedback, the system learns the partial specification(s) that best align with user preferences. These partial specifications are solved to generate more relevant recommendations.

3.2 Systems Implementation

Characteristics of the manual input, holistic, and granular systems are summarised in Figure 2. All systems are implemented in JavaScript, using the Svelte framework. Vega-lite [29] is used to render the Draco recommendations. The back-end server is implemented with Flask, and ML models are deployed using the Scikit Learn library [56].

To control for differences in user interface design, we chose not to use an existing visualization recommendation GUI in this study. However, to ensure consistency in the constraint-solving process and the recommendations generated, we adopted the Draco formal language as our recommendation generator [10] in all three recommenders (Figure 1, (3)). Draco generates visualizations based on the partial specification query. Queries can include constraints such as mark type, channels, aggregates, and data dimensions. We limit the possible values that can be included in the partial specifications to those listed in Table 1. In cases where Draco recommends visualizations that include more complex encodings (such as the row channel for trellis plots), we render the recommendation to users as-is. However, participants in the manual input system will not be able to input such encodings. Similarly, the prototype systems will not learn user preferences for these encodings. This ensures that the space of possible visualization recommendations is consistent and that the systems differ only in how users input their preferences as partial specifications.

```
"mark": "line",
"encoding": {
    "x": {
        "type": "quantitative",
        "field": "rating",
        "bin": true},
    "y": {
        "type": "quantitative",
        "aggregate": "count",
        "scale": { "zero": true } } }
```

Syntax 1: Part of a Vega-lite specification of a line chart. Only mark type and encoding information are used in our study.

```
{ "label":1, "mark_line":1,
  "x.type_quantitative":1, "x.field_minutes":0,
  "x.field_rating":1, "x.bin_True":1,
  "y.type_quantitative":1, "y.aggregate_count":1 }
```

Syntax 2: Vector representation of the same Vega-lite chart with user-provided label (1 indicates a 'liked' visualization). These are the user preferences learned by the system.

```
mark(line).

% ====== Query constraints ======
encoding(e0).:- not channel(e0, x).:- not type(e0,
     quantitative).:- not field(e0, rating).:- not
     bin(e0, 10).

encoding(e1).:- not channel(e1, y).:- not type(e1,
     quantitative).:- not aggregate(e1, count).
```

Syntax 3: The vector representation can be converted to an equivalent Draco query that includes the same information.

3.3 Holistic Binary Feedback Condition

The holistic system is a binary feedback recommender that allows users to interactively provide feedback for a visualization using binary *More like this* and *Less like this* ratings.

3.3.1 User Interface

The interface of the holistic system is divided into the main sections as shown in Figure 2. The leftmost section is the Data and Preferences panel (Figure 2A). The Data panel displays information about the data set being analyzed, including data set and dimension names. Dimension descriptions can be viewed when hovering on the information icon. The Preferences panel displays preferences learned by the system. Preferences are divided into two categories: mark preferences and encoding preferences. The opacity of the preference indicates the weight assigned to that preference by the system (i.e., how much the system thinks the user likes that preference). In the Recommendations view, four recommendations are displayed to the user (Figure 2B).

3.3.2 Interactions

Each visualization recommendation in the holistic prototype system has three buttons: *More Like This*, *Less Like This*, and a *Pin* icon. Users can indicate whether they like or dislike a visualization using the *More Like This* and *Less Like This* buttons. Visualizations can be saved using the *Pin* icon. Users can view all their pins at any time using the *Pinned*

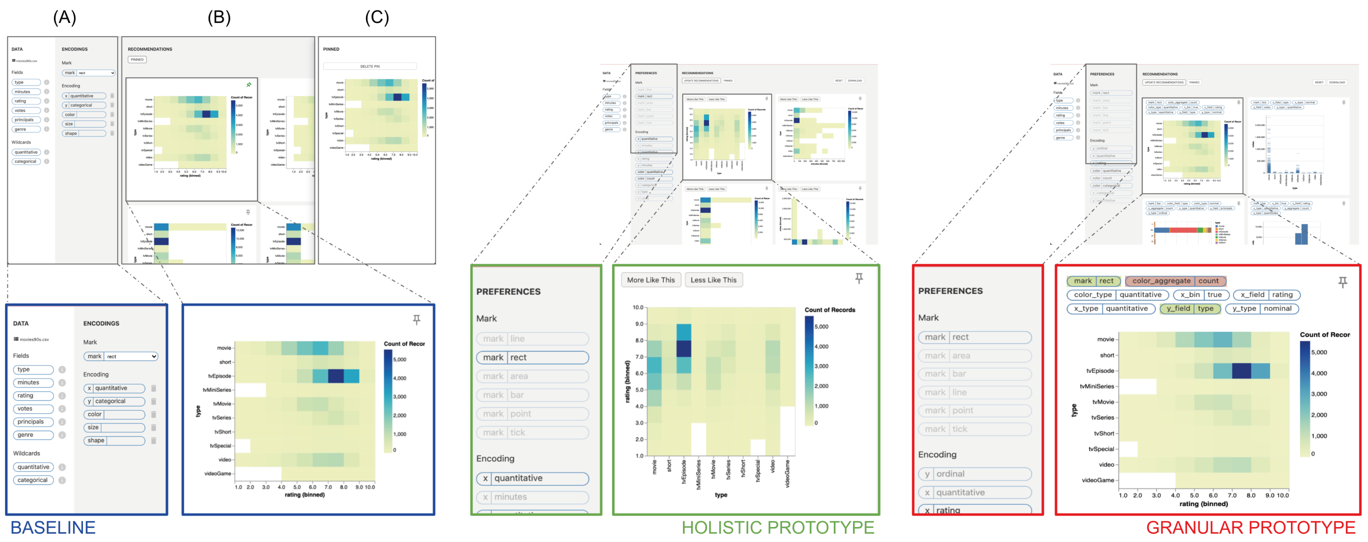

Fig. 2: Sections A, B, and C, indicated on the manual input system, make up the general layout of all three systems. A) The Data panel displays information about the data set, while the Encodings panel displays the encodings specified by the user. In the holistic and granular systems, the Encodings panel is replaced by the Preferences panel, which displays the preferences learned by the system. B) The Recommendations view displays four visualization recommendations per update. C) Users can view their pinned visualizations any time using the *Pinned* button along the top menu.

button along the top menu. They can also unpin visualizations if they change their mind about saved visualizations later in the exploration process. After viewing visualizations and providing their feedback (if any), users can refresh the recommendations using the *Update Recommendations* button.

3.3.3 Preference Learning & Recommendation Generation

Visualizations in Vega-lite JSON specification are converted into vectors by first extracting the *mark* and *encoding* features from each visualization specification (Syntax 1). We then flatten the encoding sub-fields and perform one-hot encoding on all the extracted visualization features (Syntax 2). We drop any features that are not included in our table of partial specification values (Table 1). Note that each visualization vector can then be converted into an equivalent Draco constraint query, which can then be used to generate additional instances of similar visualizations (Syntax 3).

To address the cold-start problem, the holistic system includes 50 pre-generated visualization vectors. These vectors include a range of standard visualizations, such as scatterplots, bar charts, area charts, line charts, and heatmaps. They are data set agnostic. On system load, each vector has a label of $L_i = 0$, and all 50 vectors have an equal probability of being selected. For the first set of recommendations, the system selects 4 of these pre-generated vectors, which are then converted into corresponding Draco queries (one query per vector). Four visualization solutions are requested per query (4 queries $\times$ 4 solutions $=$ 16 solutions total). One visualization is selected from each set of 4 solutions and recommended (4 recommendations total).

For each recommended visualization, users can provide feedback by clicking the *More Like This* or *Less Like This* buttons. Liked visualizations receive a label of $L_i = 1$. Disliked visualizations receive a label of $L_i = -1$. Each visualization that is labeled is converted into a vector and added to the original set of 50 pre-generated visualization vectors. Note that these visualization vectors are no longer data-agnostic (i.e., they also include information about the data dimensions used). For liked visualizations, we also add the other three visualizations generated from the same query to the pool of visualization vectors with a label of $L_i = 1$. This ensured that the system is quick to learn user likes, particularly initially, when feedback was sparse.

We use a modified PageRank algorithm to calculate the expected labels of all the vectors based on user feedback. We first train a Balltree algorithm [56] on the entire set of visualization vectors. For each vector, we query the model for its 10 nearest neighbors ($N_1, N_2..., N_{10}$), and calculate its predicted label as the average of these 10 nearest neighbors:

$$\hat{L}_i = \frac{\sum_{n=1}^{10} L_n}{10}$$

Vectors with a predicted label $\hat{L}_i > 0$ are categorized as *preferred*. Vectors with a predicted label $\hat{L}_i < 0$ are categorized as *not preferred*. Vectors with a predicted label $\hat{L}_i = 0$ are categorized as *maybe*. We then randomly select 2 *preferred* vectors and 2 *maybe* vectors[1]. Where possible, we avoid selecting all 4 vectors from the *preferred* category in order to prevent premature fixation during recommendation and exploration. The selected vectors are once again converted into equivalent Draco queries and used to generate the next iteration of recommended visualizations. To ensure that the holistic system is sensitive to changes in user preferences, the system only stores the most recent 200 visualization vectors for which the user provided feedback. Additionally, the user can clear all their labels using the *Reset* button to default to the original 50 pre-generated vectors, all with label $L_i = 0$.

1. In some cases, particularly in the initial rounds of recommendation, some categories may be empty. In this case, we would select from the next available category. Categories are always selected in the order *preferred* $>$ *maybe* $>$ *not preferred*. For implementation details, we refer you to our code repository.

3.4 Granular Binary Feedback Condition

The granular system asks users to interactively provide feedback for different features of the visualization specification, then iteratively learns user constraints and refines recommendations based on this feedback. Since the granular system is largely similar to the holistic system, we will only detail the differences in this section.

3.4.1 User Interface

The layout of the granular system is identical to the holistic system with three main sections: the Data panel, the Preferences panel, and the Recommendations view.

3.4.2 Interactions

Each recommendation in the granular system has the different features of its Vega-lite specification listed as labels above the visualization (Figure 2). Participants can click on a label once to indicate that they 'like' the feature, and once more to indicate that they 'dislike' the feature.

3.4.3 Preference Learning & Recommendation Generation

In the granular system, we convert visualizations in Vega-lite specification into vectors using the same process used in the holistic system. However, since users provide feedback for parts of the specification separately, it is possible for users to like and dislike different features of the same visualization. As such, when converting visualizations into vectors for the granular system, we drop features of the specification if no feedback was indicated, or a different feedback (e.g., 'like' instead of 'dislike') was provided. Each visualization can thus be converted into at most two vectors: one with features of the specification that were 'liked' (labeled 1) and another with 'disliked' features (labeled 0).

The recommendation generation process for the granular system is largely similar to that of the holistic system, with one minor adjustment made to the granular system in terms of prediction thresholds. Vectors with a predicted label $\hat{L}_i > 0.2$ are categorized as *preferred*. Vectors with a predicted label $\hat{L}_i < -0.2$ are categorized as *not preferred*. Vectors with a predicted label $-0.2 <= \hat{L}_i <= 0.2$ are categorized as *maybe*. Note that these thresholds are slightly different from those used in the holistic system. We had originally used 0 as the threshold, identical to the holistic system. However, one of our pilot participants felt that the system was not learning from their feedback, and some recommendations were too irrelevant. This was likely due to the fact that feedback can be relatively sparse in the granular system (e.g., a user can 'like' only the mark used in a visualization), which meant that predicted labels tended to be noisier when values were closer to 0. We thus changed the thresholds to 0.2 and -0.2 for the *preferred* and *not preferred* categories. Label prediction and recommendation selection are identical to the holistic prototype system.

3.5 Manual Input Condition

Finally, we also implemented a manual input system in order to compare binary feedback to manual inputs used in many existing visualization recommenders. The manual input system uses a shelf-configuration interaction that has been implemented in many prior tools such as Voyager [8], Voyager 2 [11], and Tableau. Users provide partial specifications that are solved to generate recommendations. We implemented our own manual input system instead of using the above tools to ensure that design factors, such as layout and color scheme, are consistent in all conditions.

3.5.1 User Interface

The interface of the manual input system is divided into the main sections as shown in Figure 2. The leftmost section is the Data and Encodings panels. This panel displays information about the data being analyzed, including dimension names. Users can create partial specifications here, which are converted into a Draco query. Four recommendations are displayed in the Recommendations view (Figure 2B).

3.5.2 Interactions

In this manual input system, users can create partial specifications by dragging data dimensions from the Data panel and dropping them into the Encodings panel. They can also specify mark type and aggregates using the respective dropdown menus. Aggregates are only available for quantitative variables. Individual encodings can be deleted using the *Bin* icon. Users can also reset all encodings using the *Reset* button along the top menu. Changes to the Encodings panel trigger a dynamic update of the system recommendations. The design of the *Pin* button is identical to the other systems.

3.5.3 Recommendation Generation

When changes are made to the Encodings panel, the new partial specification is converted into a Draco constraints query. Only four visualization solutions are requested per query. All four visualizations are then rendered in the Recommendations panel. For constraints with no returned solutions, the Recommendations panel will display the default gray background. Note that the manual input system takes partial specifications directly as provided by users. The system does not learn any user preferences.

4 User Study

In order to understand how interaction technique and interaction granularity in the different recommender systems affect data exploration outcomes, we conducted a between-subjects user study comparing the holistic and granular systems against the baseline system.

4.1 Pilot

We first recruited 4 participants for the pilot study from within the lab and through personal connections: 2 participants tested the holistic system, 1 tested the granular system, and 1 tested the manual input system. Feedback was used to make minor adjustments to the systems and the user study.

One pilot participant who had years of visualization experience used the holistic system and found it too restrictive compared to manual tools, revealing how expert users may find like/dislike-based recommendation systems limiting. We thus tailored recruitment to students at our university who were not affiliated with the visualization lab (but could be students in a visualization course). We expected this demographic of non-expert participants to better fit the intended users of like/dislike interactions.

4.2 Participants

We recruited 20 participants through mailing lists within the university and by word of mouth. The study was conducted over video conferencing software. Two participants were excluded from the final analysis due to consistently unstable internet conditions. Of the remaining 18 participants (8M, 10F), one participant was aged 35+, the rest were aged 20 to 34. We used a between-subjects study design. All participants received a $10 gift certificate as compensation.

4.3 Data sets

We used a cereals data set[2] for researcher demonstrations and an IMDB movies data set[3] for participant tasks. Different data sets were used to avoid priming effects. We chose the movies data set for the participant task because 1) we expected the data dimensions to be intuitive to participants, and 2) the data set was sufficiently complex for an open-ended exploration task. Due to the size of the movies data set, we included only English-language movies from the 90s (1990-1999). Data instances (rows) with missing dimensions were dropped. The final data set had 37788 rows. For the cereals data set, 5 dimensions and all 77 rows were used.

4.4 Tasks

Participants were asked to explore the movies data set and identify 5 interesting trends. For each interesting trend found, they were asked to pin the corresponding visualization. To ensure that participants fully engaged with the task, they were asked to describe the trends in all pinned visualizations at the end. We did not provide guidelines or requirements for findings because we wanted participants to approach this as a *bona fide* data exploration task.

4.5 Procedure

Demonstration *(10 mins)* The researcher provided a demonstration of the system the participant would be working with. The researcher used a data set about cereals that was not used by participants. Participants were then given time to ask questions about the system.
Survey *(5 mins)* The researcher provided the participant with a link to the system. The participant was given an ID and asked to provide some demographic information.
Task Description *(5 mins)* The researcher gave a description of the task (described above). Participants were given time to ask questions about the task after the demonstrations.
Task Completion *(30 mins)* Participants interacted with the system to complete the task. They were asked to think aloud while working. At the end, they were asked to review their pinned visualizations and describe the trends in each.
Debrief *(10 mins)* Participants were asked follow-up questions about their experience, such as "How did you decide to explore certain data dimensions?"

5 Results

We analyzed trends in pinned visualizations quantitatively and qualitatively. Note that while timing data was collected, unstable internet connections made fine-grained comparisons unreliable. We thus include no timing data here.

2. https://www.kaggle.com/crawford/80-cereals
3. https://www.imdb.com/interfaces/

5.1 Visualization Variety

Participants in the three conditions were compared in terms of unique marks pinned, mean number of channels per visualization, unique channels pinned, and unique data dimensions pinned. The 95% confidence intervals of the granular and holistic conditions show more overlap, while the manual input condition appears more distinct. This supports H2, which hypothesizes that the granular and holistic recommenders would be similar since they both use binary feedback. However, when we look at each measure separately, we reject H1. The manual input system resulted in lower mean mark types and encoding channels, which suggests that some specialization has occurred.

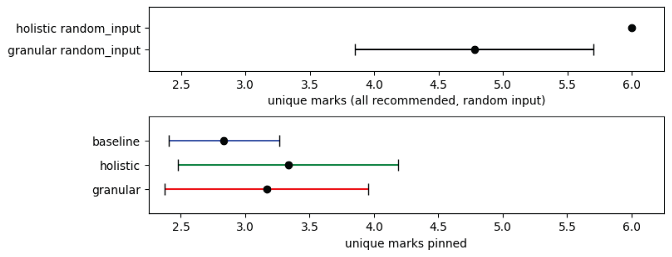

Fig. 3: Mean and 95% confidence intervals of unique marks pinned per participant. Simulation results above.

5.1.1 *Unique Mark Types Used*

Since each participant pinned five visualizations, the greatest possible number of unique marks is five – when all visualizations use a different mark – and the smallest possible number of unique marks is one – when all visualizations use the same mark. **Participants in the manual input condition explored the smallest variety of mark types,** with only a mean of 2.83 unique marks in their pinned visualizations compared to participants in the binary feedback conditions (holistic=3.33, granular=3.17, Figure 3).

5.1.2 *Encoding Channels Per Visualization*

In our study, all recommended visualizations used at least the *x* and *y* channels. Additional channels, such as *color* and *size*, could be used to encode more attributes. We analyzed channel diversity using two measures. The first measure looked at the mean number of channels used per pinned visualization. The second measure looked at the number of unique channels across all visualizations pinned by a participant (i.e., inter-visualization diversity).

Participants in the manual input condition used the least number of channels per pinned visualization, with a mean of 2.53 channels per visualization compared to participants in the binary feedback conditions (holistic=2.83, granular=2.80, Figure 4). Similarly, **manual input participants also included the smallest number of unique channels across all their pinned visualizations,** exploring only 3.33 unique channels on average compared to the binary feedback conditions (holistic=3.5, granular=3.83, Figure 4).

5.1.3 *Data Dimension Coverage*

Data dimension coverage refers to the number of data dimensions explored. We found that **participants in the**

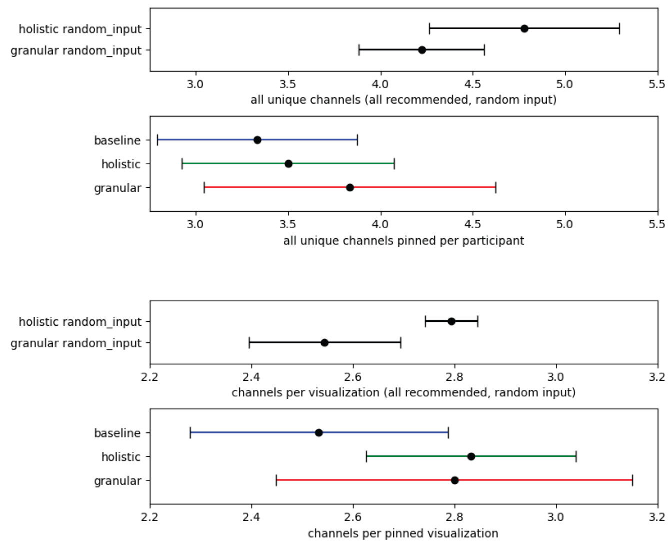

Fig. 4: Mean and 95% confidence intervals of number of channels used per visualization and total unique channels pinned per participant. Simulation results above.

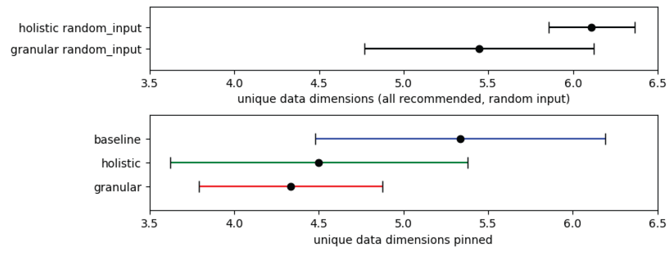

Fig. 5: Mean and 95% confidence intervals of unique data dimensions pinned per participant. Simulation results above.

manual input condition included the greatest number of dimensions in pinned visualizations, exploring a mean of 5.33 unique data dimensions compared to the binary feedback conditions (holistic=4.5, granular=4.33, Figure 5). This is consistent with prior findings that partial specification input using a shelf configuration interface led to broader data dimension coverage during exploration [8], [11].

5.2 Random-Input Simulation

To validate that the results are due to user input and not recommendation system tuning, we also ran simulations on the holistic and granular systems. Each run simulates an automated user randomly voting on visualizations. Up vote, down vote, and no feedback were equally likely. Since human participants in our study looked at an average of 32.7 (granular) and 36 (holistic) recommendations, each simulation run included 9 updates, such that a total of 36 visualizations were recommended and voted on. We made 10 runs (i.e., users) per system. We see that for most measures (unique marks, unique channels, and unique data dimensions), the simulation was more diverse than user-pinned visualizations. Since simulation results are a baseline measurement of visualization diversity, these deviations align with intuitions that human preferences are narrower and more specific than random choice. Overall, this confirms that differences between the three interaction conditions were due to user input and preferences rather than specific tuning applied to any one system.

5.3 Embedding Space

To analyze exploration patterns, we visualized the embedding space of pinned visualizations. For each pinned visualization, we took the JSON specification, dropping *$schema*, *width*, *height* and *data.url*, then flattened and dummy-encoded categorical variables. UMAP dimensionality reduction was used to create a 2D embedding space of all pinned visualizations (Figure 6).

5.3.1 Encoding Channels Coverage

The most distinct cluster in the embedding space is the holistic-granular group in the bottom left (Figure 6 V1, V10, V11, and V12). This cluster includes visualizations that map categorical variables to the color channel. Notably, there are no blue points here, suggesting that **all participants in the manual input condition failed to pin visualizations that map categorical variables to the color channel.** This under-exploration of encoding channels supports our earlier finding that participants in this condition also used the least number of channels per visualization.

We also see that the two main clusters of manual input visualizations can be characterized as bar charts about genre (Figure 6 V3) and point charts about genre (Figure 6 V9). This supports our finding that **participants in the manual input condition explored more data dimensions than participants who used other systems.** The lack of holistic and granular binary feedback recommenders in these clusters further indicates that these systems led to an under-exploration of the *genre* data dimension in particular.

5.3.2 Mark Type Coverage

Interestingly, Figure 6 did not reveal distinct clusters in terms of mark type used. Although participants in the manual input condition explored the smallest mean number of marks, there is overlap between systems (Figure 3), which may indicate a smaller difference between them.

6 Discussion

Based on our results and participant feedback, we identified two trade-offs in how different interaction techniques for partial specification support different visual data exploration strategies and outcomes. In the following sections, we discuss these trade-offs in detail.

6.1 Encodings versus Data Dimension Coverage

A recurring finding from our results is the trade-off between the number of data dimensions and the variety of mark types and channels used. The holistic and granular conditions led to more encoding channels and mark types, while the manual input condition led to more data dimensions used.

Although the difference in mark and channel coverage could be due to the greater variation in visualizations per recommendation cycle for users in the holistic and granular conditions, we found evidence to suggest that using marks and channels was also more challenging in the manual

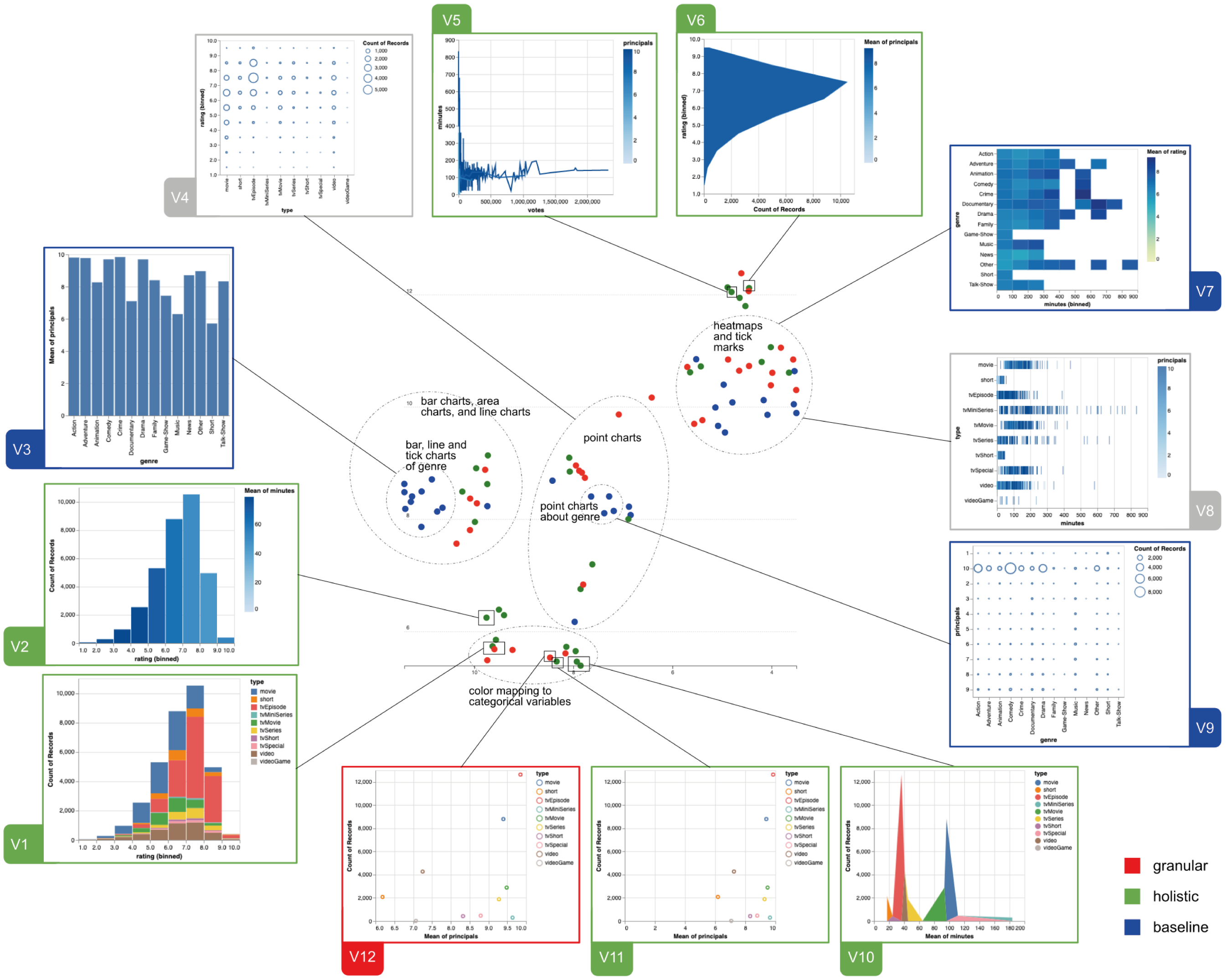

Fig. 6: The embedding space of visualizations pinned by all study participants. Visualizations from the manual input system are in blue, the holistic system in green, and the granular system in red. Regions in the embedding space are roughly denoted using dotted lines, and visualizations representative of each region have been highlighted. Visualizations outlined in gray were explored in more than one system. Interactive figure at https://gracegsy.github.io/VisRecResults/

input condition. Almost all manual input participants tried changing the mark type in the Encodings panel, but did not always know how to select from the options available. B4 decided that they were *"just going to try everything"*, while B5 switched rapidly between multiple different marks *"[j]ust to see if there's one that worked."* Some participants also avoided using more channels because they could not predict how the specification would affect recommendations. For example, B4 explained that *"I do feel like I don't know size, shape, and color... I was hesitant to use those features... If you're a pro, you probably know what these things are and you know how to get to what you want."* This sentiment was shared by B1, who said that *"I didn't [consider using color, size, and shape]. I imagine this would make things a little more complicated."* Taken together, their feedback suggests that **manual input recommenders may be better suited to users with the expertise to express their requirements as relevant queries.**

In contrast, since binary feedback recommenders automatically show users a diversity of visualizations without expecting them to first manually input partial specifications, **participants in the holistic and granular conditions could explore mark types and encoding channels they were unfamiliar with.** Furthermore, participants in the granular condition found that the interactive feedback process helped them learn about partial specifications. G3, for example, reflected that *"It just takes a few seconds to understand... the bin thing, the field thing, and the type thing."* However, these same participants in the binary feedback conditions were also less likely to use diverse data dimensions. In interviews, participants mentioned being aware of this lack of data dimension coverage. H1, for example, observed that *"it would be cool if you could see the genre here as well"*. H2 went further, saying that *"because of my 'more like this' selections, many cool things were filtered out"*. **Despite this awareness, participants in the holistic and granular conditions did not make attempts to steer recommendations towards under-explored data dimensions** or use the *Reset* button. Future studies can explore how trade-offs in data dimension coverage versus the variety of mark types and channels used might be better managed in binary feedback recommender systems.

6.2 Exploring Diversity versus Refining Specifications

Participants in our study had two main goals when interacting with the recommendation systems: 1) exploring possible visualizations, and 2) refining a specification.

Both the holistic and granular conditions helped participants explore the diversity of visualizations possible within a system, particularly in the initial stages when they may not know the visualizations and trends they wanted to see. Participants found that this helped them understand the visualization space, saying that *"I like how you have many options for users to get inspired"* (H6) and *"It's like a guided tour of the data set"* (G2). Participants also liked that the binary feedback recommendation systems helped them learn about an unfamiliar data set and the partial specification process. G1 found that *"It was a nice starting point in some ways because there was something already there. Stepping into a data set you're not familiar with, it helped me see things I might be interested in really quickly."* Similarly, G2 agreed that *"I feel like because I'm not very well versed in visualization, getting recommendations again and again, someone was showing me what I can learn about the data."* This highlights **the advantage of the binary feedback mechanisms used in the holistic and granular conditions, which allowed users to steer recommendations even without extensive domain expertise.**

In contrast to exploration, there were also cases where participants wanted to refine visualizations by making minor adjustments to the specification. Unsurprisingly, this was common in the manual input system, since recommendations were dynamically updated after each change to the partial specification. However, there were similar cases in the granular system. Most notably, when participant G4 first pinned a scatterplot (similar to Figure 6 V11), the x-axis had started at 0. G4 observed that there was overplotting, commenting that *"this graph is very crowded in this corner, making it very hard to distinguish for the x-axis"*. The participant then provided feedback before updating the recommendations. In the subsequent set of recommendations, the same scatterplot was recommended with an adjusted x-axis (Figure 6 V12). At this point, G4 said *"You made progress. This is the better one. It's not crowded anymore."* This highlights **the advantage of the manual input and granular systems, which allowed users to interact with granular components of each specification separately, such that they can more precisely indicate their preferences.**

The dual tasks of exploration and refinement confirm findings from prior studies characterizing how "open-ended tasks may decompose into focused tasks" during exploratory visual analysis [57]. Here, we supplement this with participant preferences for using certain interactions for "open-ended" exploration versus "focused" refinement. Although the granular system supported both goals, participants in this condition found it cumbersome to use the like/dislike interaction when they knew the exact changes they wanted. G1, for example, was frustrated that *"I can't request the variables that I want because I really just want the rating."* Similarly, G6 felt that *"this graph is interesting, but I want to change one thing."* This suggests that **manual input is the preferred interaction for making specific adjustments**, particularly when users have clear goals. However, by starting with ready-generated recommendations, **the granular and holistic binary feedback conditions were better for exploring the diversity of possible visualizations and learning about the partial specification process.**

7 Lessons Learned

Finally, we discuss how the interactions affect recommendation multiplicity and comprehension during visual data exploration. We also discuss the limitations of our study.

7.1 In Search of Multiplicity

Visual data exploration has often been described as a crucial step in the process of forming questions and hypotheses about data [1], where *"we rely on human pattern-finding abilities to motivate the development of future hypotheses"* [5]. Multiplicity, then, becomes a key concern during exploratory data analysis [5]. In our study, we saw that participants had a similar preference for broader coverage when completing the data exploration task. Participants in the holistic and granular conditions described how the recommendations were like *"a guided tour"*, helping them to *"get inspired"*, particularly when they were unfamiliar with the data set or with partial specification (subsection 6.2).

Perhaps more strikingly, we saw this same desire for multiplicity in the manual input condition, where participants in our study did not always input specification details intentionally. They might, for example, rapidly switch between multiple mark type options to *"try everything"* and find something that works (subsection 6.1). This trial-and-error behavior appears to be an attempt to obtain a greater variation of recommendations, suggesting that multiplicity – maximizing the diversity of visualizations seen – may be desirable during visual data exploration.

7.2 Enhancing Visualization Comprehension

While both the holistic and granular conditions led to a greater diversity of marks and channels, it is necessary to note that this diversity is not always beneficial since it adds complexity to the visualization and can be confusing, particularly for non-expert users targeted in this study. While we aimed to reduce this by asking participants to state their insight when pinning visualizations, we found that participants who used the holistic system would still pin visualizations they later could not interpret (Figure 6 V5 and V10). This indicates that the increased diversity of mark types and encoding channels seen in the holistic system did not necessarily correspond to effective use of those mark types and encoding channels. Furthermore, since none of these overly-complex (or "meaningless") visualizations appeared in the granular and manual input conditions, this suggests that just the act of interacting with the details of a visualization specification, regardless of interaction technique, may be enough to prompt users to think more deeply about visualization content.

7.3 Limitations and Future Work

Due to the technical limitations of remote testing, we had to exclude timing data in our analysis. Furthermore, our study limited partial specifications to those supported by the Draco query language. This sufficiently demonstrated the trade-offs between binary feedback and manual input, but there exist other types of specifications (such as task specifications [7], [50]) that remain to be explored. Finally, the holistic and granular systems in our study are only two possible implementations of binary feedback recommendation systems. Other implementations varying different aspects of system design may yet reveal additional trade-offs and guidelines for visualization recommenders.

8 Conclusion

This paper presents a comparative study of how non-expert users might interactively incorporate their preferences into partial specifications recommendation systems. We build and compare three visualization recommendation systems that compare binary feedback interactions at different interaction granularities to a manual input system. From the results of our study, we identify differences in data dimension coverage, mark type variety, and encoding channels used. We also provide a characterization of trade-offs and the implications of our findings on visualization multiplicity and comprehension during visual data exploration.

Acknowledgments

The authors would like to thank the Georgia Tech Visualization Lab for their insightful feedback in preparing this article. This project is supported in part by the IBM PhD Fellowship and NSF IIS-1750474.

References

[1] J. W. Tukey, “We need both exploratory and confirmatory,” *The american statistician*, vol. 34, no. 1, pp. 23–25, 1980.
[2] J. W. Tukey *et al.*, *Exploratory data analysis*. Reading, MA, 1977, vol. 2.
[3] A. Gelman, “A bayesian formulation of exploratory data analysis and goodness-of-fit testing,” *International Statistical Review*, vol. 71, no. 2, pp. 369–382, 2003.
[4] ——, “Exploratory data analysis for complex models,” *Journal of Computational and Graphical Statistics*, vol. 13, no. 4, pp. 755–779, 2004.
[5] J. Hullman and A. Gelman, “Designing for interactive exploratory data analysis requires theories of graphical inference,” *Harvard Data Science Review*, vol. 3, no. 3, 2021.
[6] A. Kale, Z. Guo, X. L. Qiao, J. Heer, and J. Hullman, “Evm: Incorporating model checking into exploratory visual analysis,” *IEEE Transactions on Visualization and Computer Graphics*, 2023.
[7] F. Bouali, A. Guettala, and G. Venturini, “Vizassist: an interactive user assistant for visual data mining,” *The Visual Computer*, vol. 32, no. 11, pp. 1447–1463, 2016.
[8] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, “Voyager: Exploratory analysis via faceted browsing of visualization recommendations,” *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 649–658, 2015.
[9] ——, “Towards a general-purpose query language for visualization recommendation,” in *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, ser. HILDA ’16. New York, NY, USA: Association for Computing Machinery, 2016.
[10] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer, “Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco,” *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 438–448, 2018.
[11] 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*, 2017, pp. 2648–2659.
[12] Z. Zeng, J. Yang, D. Moritz, J. Heer, and L. Battle, “Too many cooks: Exploring how graphical perception studies influence visualization recommendations in draco,” *IEEE Transactions on Visualization and Computer Graphics*, 2023.
[13] R. Davis, X. Pu, Y. Ding, B. D. Hall, K. Bonilla, M. Feng, M. Kay, and L. Harrison, “The risks of ranking: Revisiting graphical perception to model individual differences in visualization performance,” *IEEE Transactions on Visualization and Computer Graphics*, 2022.
[14] J. Yang, P. F. Gyarmati, Z. Zeng, and D. Moritz, “Draco 2: An extensible platform to model visualization design,” in *2023 IEEE Visualization and Visual Analytics (VIS)*. IEEE, 2023, pp. 166–170.
[15] H. Lin, D. Moritz, and J. Heer, “Dziban: Balancing agency & automation in visualization design via anchored recommendations,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–12.
[16] J. Mackinlay, P. Hanrahan, and C. Stolte, “Show me: Automatic presentation for visual analysis,” *IEEE transactions on visualization and computer graphics*, vol. 13, no. 6, pp. 1137–1144, 2007.
[17] S. Zhu, G. Sun, Q. Jiang, M. Zha, and R. Liang, “A survey on automatic infographics and visualization recommendations,” *Visual Informatics*, vol. 4, no. 3, pp. 24–40, 2020.
[18] M. Bordegoni, G. Faconti, S. Feiner, M. T. Maybury, T. Rist, S. Ruggieri, P. Trahanias, and M. Wilson, “A standard reference model for intelligent multimedia presentation systems,” *Computer standards & interfaces*, vol. 18, no. 6-7, pp. 477–496, 1997.
[19] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, “Training language models to follow instructions with human feedback,” *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.
[20] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, “Direct preference optimization: Your language model is secretly a reward model,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
[21] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan *et al.*, “Training a helpful and harmless assistant with reinforcement learning from human feedback,” *arXiv preprint arXiv:2204.05862*, 2022.
[22] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, “Fine-tuning language models from human preferences,” *arXiv preprint arXiv:1909.08593*, 2019.
[23] H. Liu, C. Sferrazza, and P. Abbeel, “Chain of hindsight aligns language models with feedback,” *arXiv preprint arXiv:2302.02676*, 2023.
[24] K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, and D. Kiela, “Kto: Model alignment as prospect theoretic optimization,” *arXiv preprint arXiv:2402.01306*, 2024.
[25] P. H. Richemond, Y. Tang, D. Guo, D. Calandriello, M. G. Azar, R. Rafailov, B. A. Pires, E. Tarassov, L. Spangher, W. Ellsworth *et al.*, “Offline regularised reinforcement learning for large language models alignment,” *arXiv preprint arXiv:2405.19107*, 2024.
[26] 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*, 2012, pp. 681–684.
[27] M. Sridevi, R. R. Rao, and M. V. Rao, “A survey on recommender system,” *International Journal of Computer Science and Information Security*, vol. 14, no. 5, p. 265, 2016.
[28] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.
[29] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, “Vega-lite: A grammar of interactive graphics,” *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 341–350, 2016.
[30] Y.-Z. Shi, H. Li, L. Ruan, and H. Qu, “Constraint representation towards precise data-driven storytelling,” in *2024 IEEE VIS Workshop on Data Storytelling in an Era of Generative AI (GEN4DS)*. IEEE, 2024, pp. 4–12.

[31] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang, "Datashot: Automatic generation of fact sheets from tabular data," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 895–905, 2019.

[32] D. Shi, X. Xu, F. Sun, Y. Shi, and N. Cao, "Calliope: Automatic visual data story generation from a spreadsheet," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

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

[34] Y. Luo, X. Qin, N. Tang, and G. Li, "Deepeye: Towards automatic data visualization," in *2018 IEEE 34th international conference on data engineering (ICDE)*. IEEE, 2018, pp. 101–112.

[35] X. Qian, R. A. Rossi, F. Du, S. Kim, E. Koh, S. Malik, T. Y. Lee, and J. Chan, "Learning to recommend visualizations from data," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1359–1369.

[36] V. Dibia and Ç. Demiralp, "Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks," *IEEE computer graphics and applications*, vol. 39, no. 5, pp. 33–46, 2019.

[37] Z. Chen, Y. Wang, Q. Wang, Y. Wang, and H. Qu, "Towards automated infographic design: Deep learning-based auto-extraction of extensible timeline," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 1, pp. 917–926, 2019.

[38] G. Wills and L. Wilkinson, "Autovis: automatic visualization," *Information Visualization*, vol. 9, no. 1, pp. 47–69, 2010.

[39] B. Saket, A. Endert, and Ç. Demiralp, "Task-based effectiveness of basic visualizations," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 7, pp. 2505–2512, 2018.

[40] S. Zhang, Y. Wang, H. Li, and H. Qu, "Adavis: Adaptive and explainable visualization recommendation for tabular data," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[41] E. Núñez Valdez, J. Cueva Lovelle, O. Sanjuán, C. Marín, and G. Hernández, "Social voting techniques: A comparison of the methods used for explicit feedback in recommendation systems," *International Joruna1 of Interactive Multimedia and Artificial Intelligence*, vol. 1, pp. 61–66, 12 2011.

[42] D. Das, L. Sahoo, and S. Datta, "A survey on recommendation system," *International Journal of Computer Applications*, vol. 160, no. 7, 2017.

[43] 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," in *Computer Graphics Forum*, vol. 41, no. 3. Wiley Online Library, 2022, pp. 145–155.

[44] D. Gotz and Z. Wen, "Behavior-driven visualization recommendation," in *Proceedings of the 14th international conference on Intelligent user interfaces*, 2009, pp. 315–324.

[45] B. Saket, H. Kim, E. T. Brown, and A. Endert, "Visualization by demonstration: An interaction paradigm for visual data exploration," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 331–340, 2016.

[46] B. Saket and A. Endert, "Demonstrational interaction for data visualization," *IEEE computer graphics and applications*, vol. 39, no. 3, pp. 67–72, 2019.

[47] K. Hu, D. Orghian, and C. Hidalgo, "Dive: A mixed-initiative system supporting integrated data exploration workflows," in *Proceedings of the workshop on human-in-the-loop data analytics*, 2018, pp. 1–7.

[48] L. Shen, E. Shen, Z. Tai, Y. Song, and J. Wang, "Taskvis: Task-oriented visualization recommendation," in *EuroVis*, 2021.

[49] 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*, vol. 28, no. 1, pp. 346–356, 2021.

[50] A. Narechania, A. Srinivasan, and J. Stasko, "Nl4dv: A toolkit for generating analytic specifications for data visualization from natural language queries," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[51] H. W. Wang, M. Gordon, L. Battle, and J. Heer, "Dracogpt: Extracting visualization design preferences from large language models," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[52] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang, Z. Tai, and J. Wang, "Towards natural language interfaces for data visualization: A survey," *IEEE transactions on visualization and computer graphics*, 2022.

[53] T. Jankun-Kelly, K.-L. Ma, and M. Gertz, "A model and framework for visualization exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 2, pp. 357–369, 2007.

[54] F. Ojo, R. A. Rossi, J. Hoffswell, S. Guo, F. Du, S. Kim, C. Xiao, and E. Koh, "Visgnn: Personalized visualization recommendationvia graph neural networks," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 2810–2818.

[55] J. Zhao, M. Fan, and M. Feng, "Chartseer: Interactive steering exploratory visual analysis with machine intelligence," *IEEE Transactions on Visualization and Computer Graphics*, 2020.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[57] 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, no. 3. Wiley Online Library, 2019, pp. 145–159.

Grace Guo is a Post-doctoral Fellow at the Visual Computing Group at Harvard University. She received her PhD in Human-centered Computing from Georgia Tech. Her current research focuses on developing visual analytics tools for explainable AI and causal inference, particularly in the domains of adult education and healthcare analytics.

Subhajit Das is an Applied Scientist at Amazon Inc. He focuses on natural language processing and causal inference models to deploy machine learning models that account for customers' preferences for delivery speed. He has a PhD in Computer Science from Georgia Institute of Technology, USA. His research investigated interactive machine learning, model optimization/selection, and designing human-in-the-loop-based visual analytic systems.

Jian Zhao is an Associate Professor in the Cheriton School of Computer Science, University of Waterloo, where he directs the WVisdom research lab. His research interests include Information Visualization, Human-Computer Interaction, Data Science, and Human-Centered AI. His work contributes to the development of advanced interactive visualizations that promote the interplay of human, machine, and data.

Alex Endert is an associate professor at the School of Interactive Computing, Georgia Tech. He directs the Visual Analytics Lab, which explores novel user interaction techniques for visual analytics. His lab often applies these fundamental advances to domains including text analysis, intelligence analysis, cyber security, manufacturing, decision making, and others.