

# Automatic Scatterplot Design Optimization for Clustering Identification

Ghulam Jilani Quadri, Jennifer Adorno Nieves, Brenton M. Wiernik and Paul Rosen

**Abstract**—Scatterplots are among the most widely used visualization techniques. Compelling scatterplot visualizations improve understanding of data by leveraging visual perception to boost awareness when performing specific visual analytic tasks. Design choices in scatterplots, such as graphical encodings or data aspects, can directly impact decision-making quality for low-level tasks like clustering. Hence, constructing frameworks that consider both the perceptions of the visual encodings and the task being performed enables optimizing visualizations to maximize efficacy. In this paper, we propose an automatic tool to optimize the design factors of scatterplots to reveal the most salient cluster structure. Our approach leverages the merge tree data structure to identify the clusters and optimize the choice of subsampling algorithm, sampling rate, marker size, and marker opacity used to generate a scatterplot image. We validate our approach with user and case studies that show it efficiently provides high-quality scatterplot designs from a large parameter space.

**Index Terms**—Scatterplot, overdraw, clustering, design optimization, perception, topological data analysis

## 1 INTRODUCTION

SCATTERPLOTS are an intuitive and widely used visualization for bivariate quantitative data that can reveal relationships and patterns between the variables [1]. Several studies have evaluated the effectiveness of scatterplots in low-level tasks [2], that include assessing trends [3], correlation perception [4], [5], average values and relative mean judgments [6], detecting outliers [7], and clustering [8], [9].

Design choices in scatterplots, including both the visual encodings, e.g., data point size or opacity, and data aspects, e.g., the number of data points, directly impact the quality of decision-making for low-level tasks [10]. Effective visualization design enhances comprehension by leveraging visual perception. Several studies have focused on optimizing a scatterplot by adjusting data point size [11], the number of data points [6], opacity [12], color [13], and shape [14].

One particular problem for scatterplots is overplotting, which occurs when many data points overlap and obscure the underlying data patterns. A combination of design choices can be made to reduce its influence, including choosing a subsampling algorithm and adjusting the sampling rate, reducing mark size or opacity, or some combination of both [15]. A designer's control over design elements that influence overplotting varies from complete control, e.g., point size and opacity, to limited control, e.g., the number of data points via subsampling, to no control, e.g., the distribution of points, which are inherent to the data. Given

the number of factors designers control, the design space is large for a manual search, as an optimal design must consider the influence each parameter has on the others and the task being performed when recommending design choices.

In this paper, we consider the problem of automatic design optimization in scatterplots for the task of *clustering*. Clustering occurs when patterns in the data form distinct groups [7], [10]. However, while identifying clustering structure is generally considered an ill-defined problem, Quadri and Rosen [8] recently introduced a model for accurately capturing human perception of different numbers of clusters in scatterplots using methods from Topological Data Analysis [8]. The method encodes the information into a *threshold plot*, calculated on the visual density to measure the visibility of different numbers of clusters in a scatterplot.

In this paper, we extend their work by utilizing the threshold plot for scatterplot design optimization. We first define a saliency measure on the threshold plot to rank the scatterplots by how salient their cluster structure is. We then evaluate an input scatterplot on the parameters that influence visual density, including data aspects, i.e., subsampling algorithm and sampling rate, and visual encodings, i.e., mark size and opacity. Finally, our approach automatically optimizes visualization designs by ranking them from highest to lowest in terms of cluster task performance.

Our approach is implemented into an open-source web tool (see Fig. 1). We validated it through a user study conducted on 70 participants from Amazon Mechanical Turk (AMT). We found that the saliency of the threshold plot is a good proxy for cluster structure when selecting an optimal scatterplot design. The effect was particularly pronounced when the value for the saliency was high. Further, a case study showed that our approach requires less interaction and time to select an optimal design over a manual search.

**Contributions:** The contributions of work are (1) an *optimization model* for parameters that ranks combinations of parameters using a saliency measure for the task of cluster analysis; (2) an open-source web tool that can be deployed to

- G.J. Quadri was with the Department of Computer Science, University of North Carolina, Chapel Hill, NC, 27599.  
E-mail: ghulamjilani@usf.edu
- J.A. Nieves was with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL, 33620.  
E-mail: jorgea1@usf.edu
- B. M. Wiernik was with the Department of Psychology, University of South Florida, Tampa, FL, 33620.  
E-mail: brenton@wiernik.org
- P. Rosen was with the Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT, 84112.  
E-mail: prosen@sci.utah.edu

Manuscript received April 19, 2005; revised August 26, 2015.

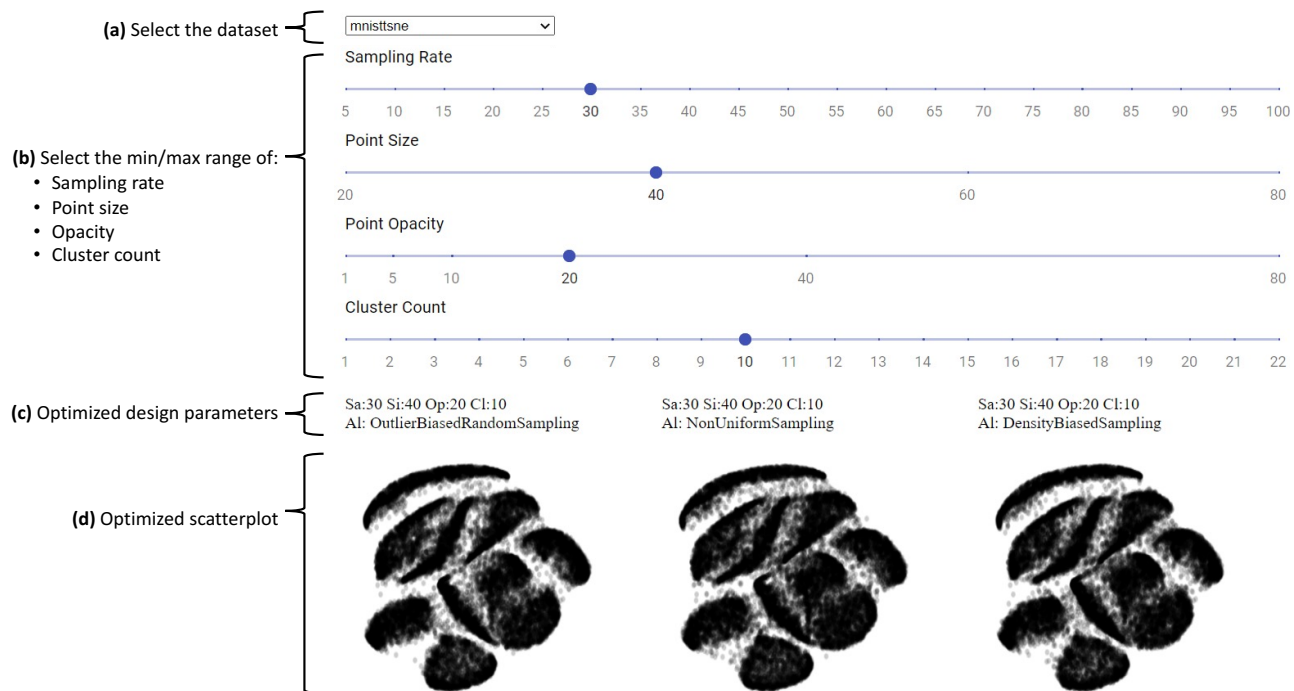


Fig. 1: Example of identifying an optimal design using our approach. (a) The user first selects a dataset. (b) Optionally, the user selects a minimum and maximum range of the parameters (in this example, the sliders are set to the same minimum/maximum values; see our demo for examples of setting ranges), including sampling rate (% of data), point size (area of circle in pixels), point opacity (% alpha), and cluster count, which the user can limit but is not optimized by our approach. (c) Our approach presents scatterplot parameters that optimize cluster saliency, ordered by saliency values 0.043847, 0.039364, and 0.021686, from left to right. (d) The scatterplots associated with the design parameters are shown.

propose optimal designs for an input scatterplot based on its clustering structure; and (3) an evaluation of the approach with a user study including 70 subjects and a case study involving 10 visualization students.

## 2 PRIOR WORK

We provide brief coverage of clustering, design optimization, and overdraw reduction.

### 2.1 Clustering in Scatterplots

Clustering, broadly defined, is the “grouping of similar data points on scatterplot in a given dataset” [10] to reveal characteristics of data and allow further exploration of underlying patterns [7], [16]. Previous works have investigated modeling cluster perception in scatterplots. Aupetit et al. studied how 1,400 variants of clustering algorithms matched human impressions of clustering structure in scatterplots and found CLIQUE, DBSCAN, and Agglomerative clustering each captured some aspects of human perception [17]. Matute et al.’s technique quantified and represented scatterplots through skeleton-based descriptors measuring scatterplot similarity [18]. However, their approach does not consider visual encodings in the evaluation. Sedlmair and Aupetit developed an approach to mimic human judgment of class separation by using machine learning on 15 class separation measures on scatterplots [19]. ScatterNet, a deep learning model, captures perceptual similarities between scatterplots that could be used to emulate human clustering decisions [20]. Scagnostics

focused on identifying patterns in scatterplots, including clusters [21], but Pandey et al. later showed they do not reliably reproduce human judgments [22].

### 2.2 Design Optimization in Scatterplots

Rensink’s framework for reasoning about perception of visualization designs suggests using techniques from vision science [23]. The extended-vision theory asserts that a viewer and visualization are a single system, whereas the optimal-reduction thesis postulates the existence of an optimal visualization. The work focuses on the fundamental question of *can we determine if a design is optimal?*

Optimization studies have focused on several aspects of scatterplots, including color assignment in scatterplot design to optimize class separability taking into account density-related factors, such as spatial relationship, density, degree of overlaps between points and cluster, and background-color [24]; creating specialized color palettes that help with visual separation of classes in multi-class data [25]; automatically selecting the optimal representation between scatterplot and line graph for trend exploration in time series data [26]; and perceptual optimization of scatterplot design on standard design parameters, including mark size, opacity, and aspect ratio, demonstrating effective choices of those variables to enhance class separation [12].

Recently, ClustMe used visual quality measures (VQMs), which algorithmically emulate human judgments, to model human perception to rank scatterplots [27]. It performed well in reproducing human decisions for cluster patterns. Their

perceptual data was later used to build a model evaluating how far existing techniques of VQM align with clusters perceived by humans [17]. In another study, 15 state-of-the-art class separation measures were evaluated, and human ground truth data on color-coded 2D scatterplots was used to learn how well a measure would predict human judgments on previously unseen data [19].

In the prior work of Quadri and Rosen, a *threshold plot* was generated using visual density of a scatterplot to model the number of clusters visible for a given set of design factors [8]. In this work, we extend the application of threshold plots to instead use them for selecting the optimal design to enhance user perception of the cluster structure in a scatterplot.

## 2.3 Overdrawing in Scatterplots and Solutions

Overplotting, the over-saturation of visual density, in scatterplots makes data analysis inefficient by obscuring underlying data patterns. A taxonomy of clutter-reduction techniques [15] suggests several approaches for reducing clutter, including varying mark size [28], [29], [30], [31], varying opacity level [32], [33], [34], [35], [36], and subsampling data points [37], [38], [39], [40].

### 2.3.1 Reducing Point Size

The size of marks in a scatterplot is an important factor in visual aggregation tasks [41]. As the size of data points on the scatterplot increases, so does the density, which directly influences cluster perception [42]. Scatterplot designs with larger points may obscure the visibility of underneath points, and hence reducing the point size would be beneficial. However, reducing the point size can conflict with color-based encodings on the data points as color difference varies with point size [13]. We consider monochrome scatterplots in our case. Therefore, we do not observe such a conflict. In the case of relatively minor overplotting, reducing point size can be helpful, but when point size is already as small as possible, i.e., 1 pixel, this method cannot be used [43].

### 2.3.2 Reducing Point Opacity

Reducing mark opacity can alleviate overplotting to assist visual analytics tasks [8], [16], e.g., spike detection in dot plots [44]. Furthermore, varying opacity levels aid in different visual tasks—while low opacity benefits density estimation for large data, it also makes locating outliers more challenging [12]. Matejka et al. defined an opacity scaling model for scatterplots based on the data distribution and crowdsourced responses to opacity scaling tasks [32]. Although a change in opacity cannot avoid overlap, it can reveal a small number of underlying or partially overlapping points or overview behavior of points [15]. Further, making the points more transparent is less helpful when there are many points.

### 2.3.3 Data Subsampling

**Sampling Rate.** The quantity of data points on the screen directly influences the visual density and overdrawing of a scatterplot. Gleicher et al.'s empirical study asked participants to compare and identify average values in multi-class scatterplots [6]. It demonstrated that judgments are improved with a higher number of points. Also, the number of data points affects the user's performance on cluster perception

in a given scatterplot [8]. Reducing the number of points reduces the overplotting and reveals underlying patterns [15].

**Sampling Algorithm.** The simplest way to reduce the number of points is to randomly sample the data, which preserves dense cluster regions but may lose low-density ones [45], [46]. Bertini and Santucci modeled the relationship between the visual density and clutter, which could be used to determine the right sampling ratio, and presented an automatic method to preserve the relative densities [47]. Improvements to the random method use non-uniform sampling that treats parts of the scatterplot differently to preserve certain properties [48]. In Sect. 3.1, we discuss several techniques that preserve relative visual density between clusters, preserve outliers when subsampling, or preserve the spatial separation between clusters.

### 2.3.4 Density-based Data Representations

There have been several variations on scatterplots that utilize alternative density representations to overcome overplotting. Carr et al. used hexagonal cells to accumulate densities [49]. Bachthaler and Weiskopf created a continuous density field using a mathematical model to produce the continuous scatterplots [50]. Keim et al. developed the generalized scatterplot, which allows users to balance overplotting and distortion [51]. Mayorga and Gleicher proposed Splatterplots, which showed dense regions as smooth contours and discrete markers to highlight outliers [52]. A recent study, called Sunspot Plots, demonstrated that a smooth blending of discrete and continuous representations enables the visualization of leading trends in dense areas while still preserving outliers in sparse regions [53].

## 3 METHODS

Visualization effectiveness is a task-dependent engagement directly impacted by the design choices. Our objective is to provide design choices for a scatterplot when optimizing for cluster structure saliency. Our approach allows interactively choosing the optimal design through a user-guided

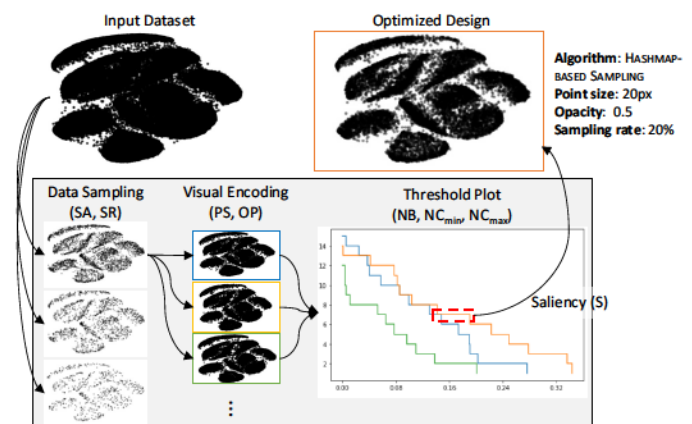


Fig. 2: Illustration of our approach. The input data (MNIST) are processed through four stages: data sampling (Sect. 3.1), visual encoding (Sect. 3.2), calculation of a threshold plot (Sect. 3.3), and finally, an optimized design is presented by ordering the saliency measure (in the red box) of scatterplots from highest to lowest (Sect. 3.4).

automatic parameterization that uses a *threshold plot* [8] to model cluster perception. The optimized parameterization of the scatterplot considers data aspects, including the number of data points, and visual encodings, including data point size and opacity. The output is a set of scatterplots ranked by their *cluster saliency* from highest to lowest.

As an overview of the process, the data are input into the following processing stages, as shown in Fig. 2.

 **Sampling** (Sect. 3.1): Data are first subsampled with different numbers of points or sampling rate using various algorithms.

 **Visual Encoding** (Sect. 3.2): A variety of data point size and opacity values are used to encode the points.

 **Threshold Plot** (Sect. 3.3): The visual density of the scatterplot is calculated and *threshold plots* are constructed.

 **Optimized Design** (Sect. 3.4): Finally, a saliency measure is extracted from the threshold plots, which are then ranked from highest to lowest saliency and presented to the user as the optimized design choice.

### 3.1 Data Sampling

 Data subsampling is dependent on sampling algorithm (SA) and sampling rate (SR). As discussed in Sect. 2.3.3, a good quality subsampling algorithm decreases the visual clutter by reducing the number of data points while retaining some of the original structure of the data. However, the best algorithms often turn out to be time-intensive to compute. Our approach considers a collection of many algorithms at a variety of sampling rates to identify an optimal one. We organize the subsampling techniques in Table 1 based on the properties they preserve, including *random*, *relative visual density preserving*, *outlier preserving*, and *spatial separation preserving*. Though all sampling rates are used by default, users may optionally select a subset of sampling rates to use.

#### 3.1.1 Random sampling

RANDOM sampling is a classic method for revealing structures in data [54]. RANDOM sampling works by selecting output samples with equal probability. Example studies using RANDOM sampling include those by Ellis and Dix [45], [46].

**Advantages & Limitations:** RANDOM sampling does not require special knowledge of the data and is widely available in existing tools. It preserves relative intensity differences, but since points are removed with equal probability, cluster structure may disappear, sampling artifacts can be introduced, and outliers may or may not be preserved.

#### 3.1.2 Preserving Relative Visual Density

The next type of sampling methods aim to preserve the visual density of both dense and sparser regions. *Visual (data) density* is the ratio between the number of displayed data samples and their corresponding rendered area. *Density preserving* algorithms optimize the weights of each sampled group to be proportional to the original group's size [55].

DENSITY BIASED sampling works by probabilistically over-sampling sparse regions and under-sampling dense regions [55], thus preserving small clusters and solitary

TABLE 1: Sampling algorithm used in our study. Category color-coding denotes the types of features preserved.

| Sampling Methods          | Application            | Category                                                                            |
|---------------------------|------------------------|-------------------------------------------------------------------------------------|
| RANDOM                    | [61], [62], [63], [64] |  |
| DENSITY BIASED            | [55], [65]             |  |
| NON-UNIFORM               | [56], [57]             |  |
| SVD                       | [58]                   |  |
| MULTI-VIEW Z-ORDER        | [40]                   |  |
| RECURSIVE SUBDIVISION     | [60]                   |  |
| OUTLIER BIASED DENSITY    | [65]                   |  |
| OUTLIER BIASED RANDOM     | [66], [67]             |  |
| HASHMAP                   | [68]                   |  |
| OUTLIER BIASED BLUE NOISE | [65]                   |  |
| BLUE NOISE                | [38], [65], [69]       |  |
| MULTI-CLASS BLUE NOISE    | [38], [70]             |  |
| FARTHEST POINT            | [71]                   |  |
| Z-ORDER                   | [40], [59]             |  |

 Random
  Preserving relative visual density
  Preserving outliers
  Preserving spatial separation

samples while reducing sampling in dense regions. NON-UNIFORM sampling strategies assign varying sampling probability to data so that some specific properties of the data can be better preserved [56], [57]. The approaches divide the sample space into a uniform grid, determines the density of each grid cell, and selects samples from cells according to their density. SVD-based sampling formulates visual density preserving as a matrix decomposition solved with singular value decomposition (SVD) [58]. This method performs SVD on the original dataset and selects the samples with the most significant correlation with top- $k$  basis vectors. MULTI-VIEW Z-ORDER sampling is a density preserving method, formulated as a set cover problem by segmenting Z-order curves of the samples in each class and the whole dataset [59]. This strategy greedily selects samples that minimize kernel density estimation error [40]. RECURSIVE SUBDIVISION sampling is a multi-class scatterplot sampling strategy to preserve relative densities, maintain outliers, and minimize visual artifacts [60]. It splits the visual space with a KD-tree and determines which class of instances should be selected at each leaf node based on a backtracking procedure.

**Advantages & Limitations:** These approaches reduce density in overdrawn regions while minimizing decreases in sparse areas. The notable feature of this category is maintaining and preserving the relativeness in the visual density of both dense and sparse regions. However, it can result in substantial cluster pattern disappearance, i.e., reduced cluster separation, and some of the algorithms are time-intensive in terms of computations, e.g., with MULTI-VIEW Z-ORDER (see Fig. 6).

#### 3.1.3 Preserving Outliers

Preserving outliers is another general goal in sampling strategies. Having no clear definitions, data points in low-density areas are often regarded as outliers [72]. A typical method for achieving this goal is to update existing sampling algorithms to make them accept more outliers [65], [66].

OUTLIER BIASED RANDOM sampling assigns higher sampling probabilities to outliers in random sampling [66]. Other sampling methods have also been adapted to bias their

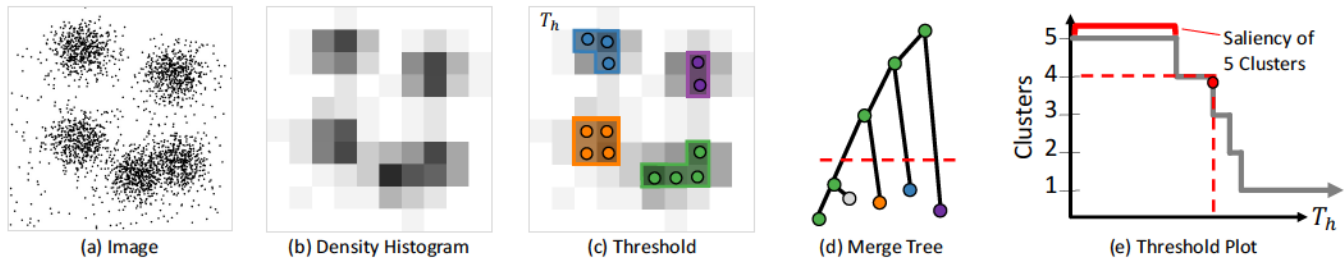


Fig. 3: Illustration of generating a *threshold plot* (refer to [8] for details) to compute a saliency score. (a) A scatterplot is given as input. (b) A density histogram (of  $20 \times 20$  in our implementation) is calculated. (c) The histogram is evaluated at different density thresholds, and components are extracted. (d) A merge tree is created by tracking components across thresholds. (e) A threshold plot is generated from the merge tree. The horizontal axis represents the a persistence threshold of clusters, while the vertical axis shows the number of clusters visible at that threshold. The dashed red line shows how a threshold can be extracted from a given number of clusters and vice versa. Finally, the saliency for some number of clusters is measured as the range of the minimum and maximum threshold for that number of clusters.

sampling towards outliers, e.g., OUTLIER BIASED BLUE NOISE sampling [66] and OUTLIER BIASED DENSITY sampling [65]. HASHMAP-based stratified sampling technique preserves outliers while keeping the main distribution by sampling the point clouds on display using a color mapping [68].

**Advantages & Limitations:** Preserving outliers conflicts with many of the goals of emphasizing cluster separation. For example, preserving outliers will distort the relative data densities since relatively more data points are selected in low-density regions instead of high-density ones, which will increase the ambiguity between cluster boundaries.

### 3.1.4 Preserving Spatial Separation

There are some cases where spatial separation between classes/clusters or highly dense regions is desirable.

BLUE NOISE sampling, inspired by [73], randomly selects samples but remains spatially uniform [69], [74]. MULTI-CLASS BLUE NOISE is a multi-class extension that maintains the blue noise properties of each class and of the whole dataset [70]. FARTHEST POINT sampling selects samples with better spatial separation by randomly selecting the initial sample and iteratively selecting additional samples of maximal minimum distances to previous ones [71]. Z-ORDER-based sampling uses space-filling curves to sample [40].

**Advantages & Limitations:** This category of method maintains spatial distribution and separation, which helps in identify underlying clustering patterns. However, the algorithms are time-intensive, and sampling computation time increases with the number of data points, e.g., BLUE NOISE sampling (see Fig. 6).

## 3.2 Visual Encoding

Once data are subsampled, they are rendered multiple times, varying several visual encodings. Prior studies have demonstrated the effect of visual encodings on analysis tasks [13], [75], [76], and visual encodings influence group or separation perception [77], such as, color, size, shape [14], orientation [78], texture [79], opacity [12], density [80], motion and animation [81], [82], [83], chart size [84], and others. Additionally, studies have demonstrated a perceptual effect in scatterplots with changes in the factors, including data distribution types, number of points, the

proximity of concentrations of points, data point opacity, and relative density [6], [11], [13], [42], [44], [75], [85]. Visual encoding is dependent on point size (PS) and point opacity (OP). By default, users are provided a pre-selected set of values for these parameters (see Sect. 4), but they may limit them to a subset, if desired.

## 3.3 Threshold Plot: Computation of Saliency Score

Next, we take the generated scatterplots and compute *threshold plots* and a saliency score. The threshold plot is a monotonic step function, where the horizontal axis encodes values that describe the separation of clusters, while the vertical axis describes the number of clusters visible at that threshold. We extract from this plot the number of clusters an individual is likely to see and exactly how salient those clusters are.

### 3.3.1 Merge Tree Model of Visual Density

We utilize the visual density-based model, first introduced by Quadri and Rosen [8], which attempts to directly identify the *relative* visual density, i.e., the number of filled pixels, at which users will differentiate between clusters. They showed that this visual density-based model was a good proxy for predicting the number of clusters a human would perceive in a scatterplot. In contrast, for this paper we are trying to show that this same model can be used for design optimizations of scatterplots. We briefly summarize their approach.

The model first encodes the clustering structure as a function of density using a merge tree. The merge tree is a data structure from Topological Data Analysis that encodes the merging order of sublevel sets of the visual density. As shown in Fig. 3, the basic process is: (a) an input scatterplot *image* (integrating all the design factors, including SA, SR, PS, and OP) has its (b) density histogram calculated at a pre-determined size of  $20 \times 20$ , as proposed by [8]. (c) For a given density value,  $t$ ,  $x$  number of clusters are observed using the 8-connected neighbors. (d) The merge tree tracks the appearance and merging of clusters (i.e., when clusters blend to be perceived as one) across all density values,  $t: 0 \rightarrow \infty$ . The merge tree is efficiently calculated using the join tree of a scalar field (see [86] for an efficient algorithm).

The next step is that for each cluster identified in the merge tree, the persistence [87] of that cluster,  $\rho$ , is calculated

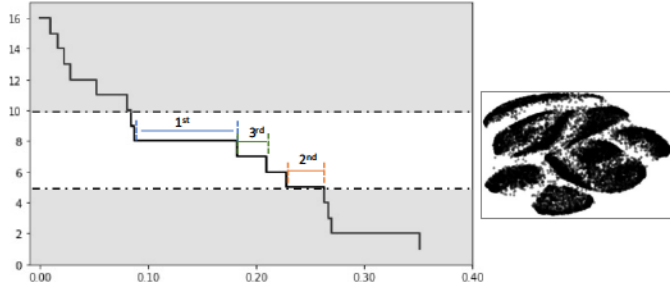


Fig. 4: The threshold plot shows the persistence on the x-axis and number of clusters on the y-axis. The longer the bar (i.e., the saliency), the more visible the clustering structure. Within a user-selected range (white ribbon) of 5-10 clusters, the three prominent bars — — — are highlighted, but — is the most salient.

by considering the difference between the highest ( $t_h$ ) and lowest ( $t_l$ ) density values where that cluster is visible, i.e.,  $\rho = t_h - t_l$ . The fundamental intuition behind persistence is that it measures the relative scale of a feature (e.g., the relative change in density), as opposed to the absolute scale of the feature (e.g., the absolute density value). (e) The threshold plot encodes for a given threshold, exactly how many clusters have a persistence greater than or equal to it.

### 3.3.2 Scatterplot Cluster Saliency

Using the threshold plots as-is would represent an under-constrained optimization, as it would require, at the very least, a user specification of the number of clusters or a persistence threshold.

Therefore, we wish to optimize by maximizing the *dynamic range* or *saliency* in the threshold plot. For a given number of clusters,  $C_i$ , the saliency is  $T_{i,max} - T_{i,min}$  (see Fig. 3(e)). We represent the saliency of the scatterplot by the length of the bar with the maximum individual saliency. In other words, the largest saliency is considered the best representation of the clustering structure of the scatterplot. By default all cluster counts are considered. However, users may also limit which bars are considered by selecting a range,  $[C_{min}, C_{max}]$ , for the number of clusters of interest. In Fig. 4 there are three prominent threshold bars — — —, in the range of 5 – 10 clusters, but the bar — represents the most salient clustering structure in the scatterplot.

## 3.4 Optimized Design



To enable selecting the best design, the saliency (i.e., maximum bar lengths) are used to directly compare scatterplots. In Fig. 5, the bar length of the blue scatterplot indicates that it has more salient clustering structure than the scatterplot in green, a quality that can also be observed in the scatterplots themselves.

Finding the optimal scatterplot is done by first rendering all combination of SA, SR, PS, and OP (from their minimum to maximum user defined ranges). The threshold plots and scatterplot saliency are calculated. The optimized scatterplot design is ranked using saliency score by selecting:

- SA among the finite set of sampling algorithms in Sect. 3.1;

- SR among the finite set  $\{SR_{min}, \dots, SR_{max}\}$ ;
- PS among the finite set  $\{PS_{min}, \dots, PS_{max}\}$ ;
- OP among the finite set  $\{OP_{min}, \dots, OP_{max}\}$ ; and
- $C_{min}$  and  $C_{max}$  are user-selected but not optimized.

Finally, all scatterplots are ranked from most salient to least salient and provided to the user.

## 4 USER-GUIDED OPTIMIZATION INTERFACE

We developed an interactive web interface to demonstrate our approach (which is used in Sect. 7), as seen in Fig. 1, where one can select an optimized scatterplot based on the cluster structure saliency. The interface enables optimizing visual encodings, i.e., point size (PS) and opacity (OP), and data aspects, i.e., subsampling algorithm (SA) and sampling rate (SR), on real-world data using the saliency of threshold plots. The user can select the ranges for parameters ( $PS_{min}, PS_{max}, OP_{min}, OP_{max}, SR_{min}, SR_{max}$ ) and cluster count ( $C_{min}, C_{max}$ ) for more refined ranking. Here, we detail the stages illustrated in the overview from Fig. 2.

**Operation of the Interface.** Our interface outputs the ranking of scatterplots by their cluster structure saliency, also known as *saliency score*. The user selects the dataset and can optionally choose different ranges for sampling rate, point size, opacity, and cluster count. The output ranks and presents the scatterplots with the highest saliency.

**Input Datasets.** We selected datasets from the prior studies in visualization as our experimental data. We selected eight representative datasets (see Fig. 15) with different characteristics: six datasets with clustering structures (MNIST ( $n = 70000$ ) [88], Conditional Based Maintenance ( $n = 10000$ ) [89], Clothes ( $n = 26569$ ) [61], Crowd-sourced Mapping ( $n = 10845$ ) [90], Epileptic Seizure ( $n = 11500$ ) [91], Swiss Roll 2D ( $n = 8000$ ) [92]) and two with curved stripes (Swiss Roll 3D ( $n = 10000$ ) [92] and Abalone ( $n = 4177$ ) [93]). Four additional examples appear only in our demo application: census-income ( $n = 48842$ ) [94], pp-gas-emission-2011 ( $n = 36733$ ) [95], credit-card ( $n = 30000$ ) [96], diabetes-data ( $n = 50000$ , only half of the dataset) [97]. For datasets with dimensionality higher than two, we first transformed them into 2D data using t-SNE and normalized them to  $[0, 1] \times [0, 1]$ .

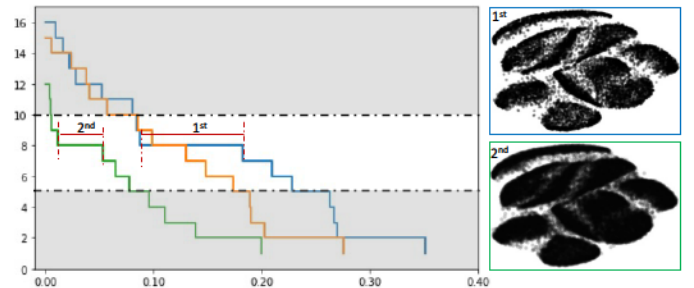


Fig. 5: By ranking the saliency of scatterplots, we identify the one with the clearest cluster structure. In this example, there are two prominent bars within the user-defined range of 5-10 clusters (white ribbon) — —, but — represents the clearer cluster structure.

**Subsampling (SA and SR).** We dataset using the 14 algorithms (SA). To understand the performance of technique, we selected sampling rates (SR) [5%, 95%] of the input data with a step size (Visual Encoding (PS and OP). Data as point marks (i.e., circles ●) on the two visual encodings that influence are varied. The point size (PS) is selected: {20<sub>px</sub>, 40<sub>px</sub>, 60<sub>px</sub>, 80<sub>px</sub>}, and the point opacity to be {1%, 5%, 10%, 20%, 40%, 80%}. Both r. sizes are selected using guidelines from [8].

**Scatterplot Rendering.** For each dataset, s rendered using all combinations of  $SA \times SR$  default, they are rendered with image dim [700<sub>px</sub> × 700<sub>px</sub>], which was selected such would fit vertically on the majority of desktop monitors without scrolling [98] with a horizontal resolution to match. Any data falling outside this region is clipped.

**Saliency Computation.** The fundamental unit our approach is the *visual density*, in particular the point at which human perception of the visual density of cluster distributions will blend to be perceived as one. For each scatterplot generated in the prior steps, a threshold plot is generated, and the cluster structure saliency of the scatterplot is calculated.

**Optimized Design.** The final results are the ranked order of scatterplot designs based on their saliency. One point to be noted here is that many scatterplot designs produce similar saliency values because they are perceptually similar (refer to Sect. 6 for more details).

## 5 QUANTITATIVE ANALYSES

To understand the operational aspects of our approach, we performed the following analyses on eight datasets.

### 5.1 Computation Time

**Subsampling.** We recorded computation time for subsampling on every dataset, on each algorithm, and for each sampling rate. We observed similar patterns for all datasets. Therefore, we will discuss the results for only MNIST. Fig. 6 shows the results. The approaches roughly break down into three groups with low (e.g., RANDOM and OUTLIER

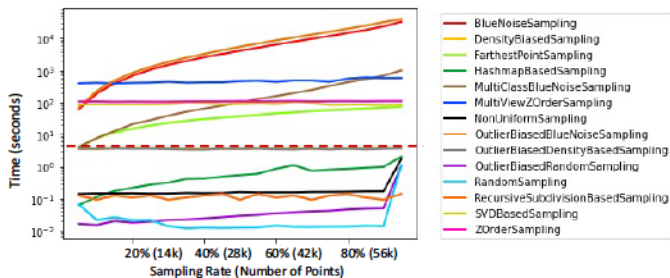


Fig. 6: Subsampling computation time (logarithmic scale) for 14 algorithms across different sampling rates (linear scale, 5% to 95% with interval of 5%) for the MNIST dataset. The dashed red line - - - represents the time needed to render an image (see Fig. 7).

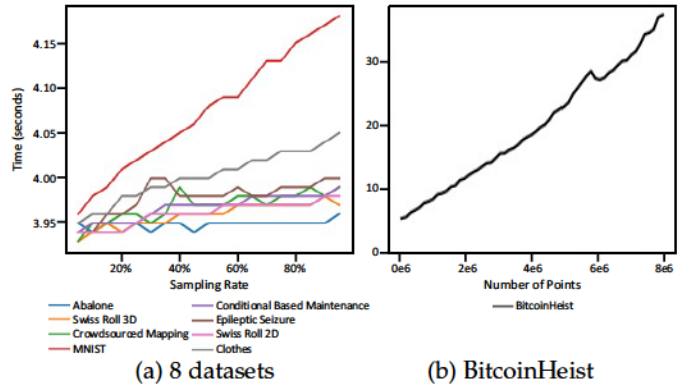


Fig. 7: Rendering and saliency computation time for (a) 8 datasets used in all analysis and (b) *BitcoinHeist* [99] dataset used for scalability. Each point is the average time for 14 (sampling algorithms) × 4 (point sizes) × 5 (point opacities). Sampling computation time (linear scale) is dependent on number of points (linear scale), i.e., the sampling rate.

BIASED sampling), medium (e.g., SVD-BASED and FARTHEST POINT sampling), and high (e.g., BLUE NOISE and OUTLIER BIASED BLUE NOISE sampling) completion time. The second observation is that some algorithms (e.g., NON-UNIFORM, OUTLIER BIASED DENSITY, and RANDOM sampling) perform uniformly for all sampling rates, while other algorithms (e.g., BLUE NOISE and OUTLIER BIASED BLUE NOISE) have completion times that increase as the sampling rate increases.

**Rendering and Saliency Extraction.** After subsampling the data, the scatterplot is rendered, the threshold plot is calculated, and the saliency is extracted. We computed and recorded the average time taken for each dataset (excluding subsampling) in Fig. 7a. The computation time fits in a relatively small window around 4 seconds, though a general trend shows that the computation time is proportional to the number of data points, probably owed to increased rendering costs. Furthermore, the dashed red line in Fig. 6 shows the average time for the rendering and saliency computation time of approximately 4 seconds. While several subsampling methods take less time, many require significantly more. Hence, there is a trade off between time and quality, which we explore in the next section.

**Scalability.** To further analyze the computation for more data points in terms of scalability, we selected a dataset, *BitcoinHeist* [99], with approximately 3 million data points. We computed and recorded the computation time for rendering and saliency calculations. The trend seen in Fig. 7b demonstrates the linear characteristic of computation time with number of data points.

### 5.2 Subsampling Quality

The computation of subsampling is a significant portion of processing time. An important question to reflect upon is whether all of the subsampling methods are necessary, particularly those requiring high computation time, e.g., in Fig. 6, BLUE NOISE and OUTLIER BIASED BLUE NOISE sampling take several orders of magnitude more compute time compared to RANDOM sampling or OUTLIER BIASED RANDOM sampling.

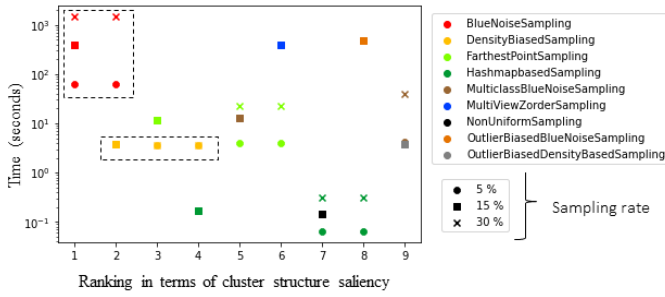


Fig. 8: Evaluation of SA time computation vs. performance on three sampling rates (5%, 15%, and 30%) on the MNIST dataset. The shape marker represents the SR, and the color represents the SA. The chart shows computation time and the rank of the top nine sampling methods. The top left corner shows BLUE NOISE sampling as the top ranked method, but it also required more computation time.

We performed a comparative analysis of the algorithms by selecting the MNIST dataset with a sampling rate between 5% to 30% and looked at the top-ranked methods. Fig. 8 shows the evaluation of 14 SA time computations against their performance by measuring how frequently the SA produces the optimal scatterplot design. We included top nine SAs at three pre-selected SRs in the figure.

We found that BLUE NOISE and DENSITY BIASED sampling methods are the top two ranked algorithms, followed by FARTHEST POINT sampling as the third in line (see Fig. 8). The main reason behind this ranking is the feature preservation, i.e., spatial separation (see Table 1). From the top two methods, we found that there is a higher computation time for BLUE NOISE than DENSITY BIASED sampling methods. However, BLUE NOISE generated more salient structure, while DENSITY BIASED produced slightly less salient structure but also took less time comparatively. The important conclusion here is that some techniques that take longer to compute can provide the best results.

## 6 USER STUDY

To validate the utility of the threshold plot saliency for ranking scatterplots based on their clustering structure, we ran a user study on Amazon Mechanical Turk (AMT).

### 6.1 Study Design

#### 6.1.1 Hypotheses

**[H1]** *Similar patterned threshold plots represent scatterplots that are perceptually similar and have similar cluster structures.*

We believe that threshold plots can be used as a proxy to identify which scatterplot designs have more salient structure, and scatterplots with similar threshold plot shapes have similar visual density and visual separation.

**[H2]** *The longer the maximum threshold bar, the more salient the cluster structure is in a scatterplot.*

We further consider the length of the longest bar, i.e., *saliency*, to be a valid feature for ranking scatterplot designs.

#### 6.1.2 Study Task

We utilize two tasks, [T1] and [T2], for [H1] and [H2], respectively.

**[T1]** *Which scatterplot has more similar cluster structure to the reference scatterplot?*

A reference and two other scatterplot designs are shown, and subjects have to select the scatterplot design with the more similar cluster structure to the reference plot.

**[T2]** *Which scatterplot has a clearer cluster structure?*

Two scatterplots are shown, and subjects have to select the one with a clearer cluster structure. Each scatterplot has a calculated saliency value, and those with a higher saliency value should have a clearer cluster structure.

### 6.2 Stimulus Generation

**Data Selection.** We selected six datasets (MNIST, Conditional Based Maintenance, Clothes, Epileptic Seizure, Swiss Roll 2D, and Swiss Roll 3D) from those listed in Sect. 4. In addition, Crowdsourced Mapping was used for the training examples, and Abalone was excluded for having a similar shape in the scatterplot to the Swiss Roll 3D.

**Scatterplot Rendering.** The scatterplot images are rendered with similar parameters as those in the interface.

- Stimuli dimensions ( $[X \times Y]$ ):  $[700_{px} \times 700_{px}]$
- Data point size/area (PS):  $\{20_{px}, 40_{px}, 60_{px}, 80_{px}\}$
- Data point opacity (OP):  $\{1\%, 5\%, 10\%, 20\%, 40\%, 80\%\}$
- Sampling rate (as a proportion of the number of data points) (SR): On the interval  $[5\%, 95\%]$  with 5% step size using all 14 SA techniques from Table 1.

### 6.3 Study Procedure

#### 6.3.1 [H1] Threshold Plot Difference as Perceptual Similarity

Two similar scatterplots potentially represent similar cluster structures, and it can become ambiguous to distinguish between them. In our approach, two scatterplots are *perceptually similar* if their threshold plots are similar to each other (e.g., see Fig. 9). As a metric to determine the perceptual similarity between clustering structures, we use the area under the curve (AUC), i.e.,  $AUC(X) = \sum_{i=1}^n |x_i|$  for a threshold plot, as a measure to compare between scatterplots.

We calculated the distribution of AUCs for threshold plots in the MNIST datasets (see Fig. 10). The AUCs are divided

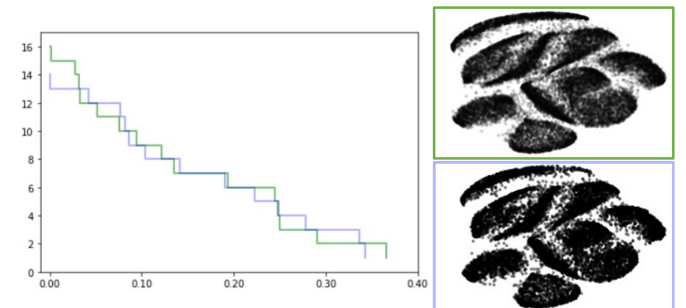


Fig. 9: The threshold plots (left) show similar patterns in the clustering structure of their associated scatterplots (right).

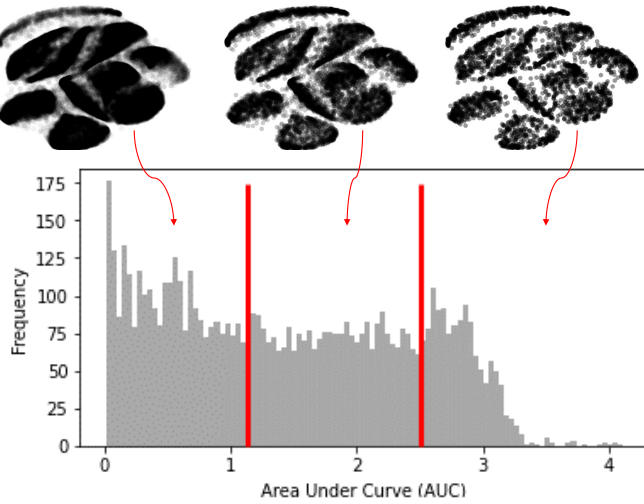


Fig. 10: Histogram of Area Under Curve (AUC) for MNIST with example scatterplots.

into three equally sized bins. Finally, we use three similarity criteria: Similar (SR) if scatterplots are from the same bin; Somewhat Similar (SS) if scatterplots are from adjacent bins, e.g., 1st and 2nd, or 2nd and 3rd; and Dissimilar (DS) if scatterplots are from the 1st and 3rd bin.

### 6.3.2 [H2] Threshold Plot Bar Length as Saliency

As we consider optimizing the saliency of plots, one question naturally occurs, which is how are the saliency values distributed across the parameters we have selected. While the precise distribution is data-dependent, all fall into a similar trend that can be observed for the MNIST dataset in Fig. 11. The vast majority of configurations lead to low cluster saliency, and few configurations provide the optimal saliency, which makes finding that optimal saliency by a manual search (i.e., manually selecting parameters), instead of our approach, difficult.

For our analysis, we divide the space of saliency values for a given dataset into three evenly spaced groups: low, medium, and high saliency. In the example of Fig. 11, the bins are: low  $[0.0, 0.033]$ , medium  $[0.033, 0.067]$ , and high  $[0.067, 0.1]$ .

### 6.3.3 Stimulus and Trials

We keep the number of trials small ( $6 \times [T1]$  and  $12 \times [T2]$ ) for both tasks to reduce the risk of learning effects. The scatterplots for stimuli are selected from the generated pool (number of datasets (D)  $\times$  SA  $\times$  SR  $\times$  PS  $\times$  OP) in random order but in the following manner:

For [T1], the subject is shown three scatterplots in one stimulus: one reference (R) and two as a forced-choice. The reference (R) is displayed above and two options are shown below. From the three bins (B) of AUC perceptual similarity scatterplots are classified as: similar (SR), somewhat similar (SS), and dissimilar (DS), with respect to the reference scatterplot, R. For two scatterplots, A and B, the possible options are:

- 1) A is similar to R. B is dissimilar to R.
- 2) A is similar to R. B is somewhat similar to R.
- 3) A is somewhat similar to R. B is dissimilar to R.

Next, we have these three combinations for the [T1]:  $SR \times R \times DS$ ,  $SR \times R \times SS$ , and  $SS \times R \times DS$ . Each trial randomly selects one combination from the above options for each dataset:  $D(6) \times B(1) = 6$  stimuli for [T1].

For [T2], the subject is shown two scatterplots for one dataset as a forced-choice with saliency values divided into three bins (B) of saliency, High (H), Medium (M), and Low (L). Next, we have these six combinations for the [T2]:  $H \times H$ ,  $H \times M$ ,  $H \times L$ ,  $M \times L$ ,  $M \times M$ ,  $L \times L$ . Each trial randomly selects two combinations among those six for each dataset:  $D(6) \times B(2) = 12$  stimuli for [T2].

### 6.3.4 Interface

We used a webpage for the experiments, where each participant was given 18 stimuli ( $6 \times [T1]$  and  $12 \times [T2]$ ) selected randomly from the generated pool. The maximum allocated time, which was visible to participants, for one trial of [T1] was 15 seconds, and [T2] was 10 seconds. At the expiration of time, the page was automatically advanced. At the beginning of the experiment, we included a brief introduction, examples, and one training task per task type using the Crowdsourced Mapping dataset. We also included an open ended post-test questionnaire for general feedback on usability and quality. The experiment was expected to last less than 10 minutes in total. The consent was obtained at the beginning of the experiment. Please refer to the user study demo at <http://scatter.projects.jadorno.com/>.

### 6.3.5 Participants

We recruited participants from Amazon Mechanical Turk (AMT) for the IRB-approved study. Based upon a post hoc power analysis of the preliminary experiment data, we recruited a total of 70 participants (49 male, 21 female; ages:  $[18 - 64]$ , median age group:  $[25 - 44]$ ) limited to the US or Canada. 47% of participants reported having corrected vision. All participants had a HIT approval rate of  $\geq 95\%$  and were compensated at US Federal minimum wage or above.

In total, task [T1]: 6 trials  $\times$  70 participants = 420 responses, and task [T2]: 12 trials  $\times$  70 participants = 840 responses, were collected. We carried out some data quality

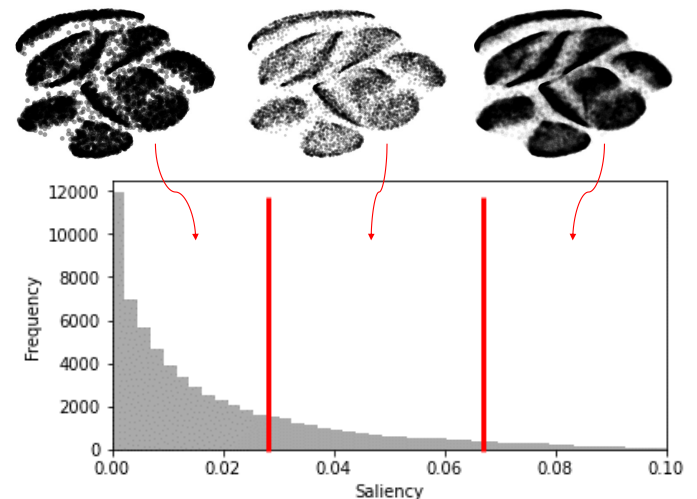


Fig. 11: Histogram of saliency for all MNIST scatterplots with example scatterplots.

checks on responses with the constraints: participant responses with timing less than one second for a given trial were rejected, and trials with no response or expired time were rejected. We identified one trial for task [T2] with no response from the subject and hence rejected, leaving a total of 839 responses for analysis.

## 6.4 Analysis Methodology

### 6.4.1 Model Specifications

To test our hypotheses, we fit models predicting whether subjects would choose the theoretically-predicted ("target") response option rather than the alternative ("comparison") option. This was a binary choice, so we modeled the choice using Bernoulli regression models. A Bernoulli regression is a generalized linear model that estimates the probability of an event (i.e., the subject choosing the target option) occurring using a linear combination of predictors. To ensure only valid probabilities in the  $[0,1]$  range were estimated, we used the logit (log-odds) link function. In each model, our focal predictors were the proxy indicators for the visualization characteristic hypothesized to determine choice in our theoretical model (i.e., AUC for [T1], saliency/bar length for [T2]) for the two response options.

Support for our hypotheses would be indicated if the proxy indicators strongly and significantly predicted subject choice of the target option. When there is a large difference between proxy indicator values for the two options, subjects should select the target option with high probability. However, when the two options have similar proxy indicator values, then subjects should select the two options at similar rates (i.e., select the target option with probability  $\approx .50$ ).

**Task 1.** In [T1], we predicted subjects would choose the scatterplot with the smaller AUC from the reference plot as being more similar in cluster structure. AUC captures how similar a plot is to the reference plot. Small AUC differences indicate that the plot is highly similar to the reference plot. Large AUC differences indicate that the plot is highly dissimilar to the reference plot. We predicted that the probability of choosing the target option should increase the larger the difference in AUC values between the two response options. That is, when the target is much more similar to the reference plot than is the comparison plot, participants should consistently choose the target plot. When the two plots are equally similar (or dissimilar) to the reference plot, participants should select each at similar rates. As described above, we modeled that probability that the subject would choose the target (more-similar) option in trial  $t$  using the AUC values for the target (TAUC) and comparison (less-similar; CAUC) options and included random intercepts for each respondent ( $i$ ) and stimulus dataset ( $j$ ) to account for dependency across trials:

$$\begin{aligned} \text{ChooseTarget}_{ijt} &\sim \text{Bernoulli}(p_{ijt}) \\ \text{logit}(p_{ijt}) &= \beta_1 \times \text{TAUC}_{ijt} + \beta_2 \times \text{CAUC}_{ijt} + \\ &\quad \alpha + \gamma_i + \delta_j \\ \gamma_i &\sim \text{Normal}(\mu_\gamma, \sigma_\gamma) \\ \delta_j &\sim \text{Normal}(\mu_\delta, \sigma_\delta) \end{aligned} \quad (1)$$

TABLE 2: Model predictions for [T1] show the predicted probabilities of choosing the target option (Pr(CT)) based on the fitted model, with 95% CI, for combinations of target and comparison AUC (see Sect. 6.3.1) values. For SR, SS, and DS, see Sect. 6.3.3. SE=standard error, CI=confidence interval.

| Target [AUC] | Comp. [AUC] | AUC diff. | Pr(CT) | SE   | 95% CI       |
|--------------|-------------|-----------|--------|------|--------------|
| SR [0.0]     | SR [0.0]    | 0.00      | 0.61   | 0.09 | [0.43, 0.77] |
| SS [1.0]     | SS [1.0]    | 0.00      | 0.64   | 0.08 | [0.48, 0.78] |
| DS [2.0]     | DS [2.0]    | 0.00      | 0.67   | 0.11 | [0.44, 0.84] |
| SR [0.0]     | SS [1.0]    | 1.00      | 0.75   | 0.06 | [0.62, 0.85] |
| SS [1.0]     | DS [2.0]    | 1.00      | 0.77   | 0.05 | [0.65, 0.86] |
| DS [2.0]     | DS [3.0]    | 1.00      | 0.79   | 0.08 | [0.60, 0.91] |
| SR [0.0]     | DS [2.0]    | 2.00      | 0.85   | 0.05 | [0.74, 0.92] |
| SS [1.0]     | DS [3.0]    | 2.00      | 0.86   | 0.04 | [0.75, 0.93] |
| DS [2.0]     | DS [4.0]    | 2.00      | 0.88   | 0.06 | [0.71, 0.96] |
| SR [0.0]     | DS [3.0]    | 3.00      | 0.91   | 0.04 | [0.81, 0.96] |
| SS [1.0]     | DS [4.0]    | 3.00      | 0.92   | 0.04 | [0.82, 0.97] |
| SR [0.0]     | DS [4.0]    | 4.00      | 0.95   | 0.03 | [0.85, 0.99] |

Pr(ChooseTarget) by Difference in AUC from Reference

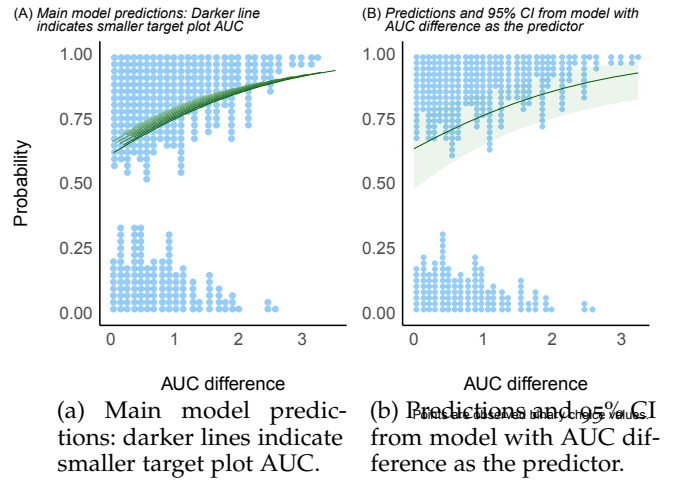


Fig. 12: User study results for [T1], Pr(ChooseTarget) by AUC difference from reference. The blue dot represents data points for each observation with lines showing predicted probabilities of choosing the target option as a function of difference between the AUC values for the two stimuli options. Panel (a) shows results for the main model, where each line reflects a combination of Target AUC and Comp. AUC as shown in Table 2. Darker lines indicate higher Target AUC. The results show that as the AUC difference increases, probability of choosing the target option increases strongly, and there is little variation in predicted probability of choosing the target option by levels of Target AUC (different lines). Panel (b) shows results for an alternative model using AUC difference as the sole predictor. The single line shows predicted probability of choosing the target option by levels of AUC difference, with 95% confidence band. The results show that as the AUC difference increases, probability of choosing the target option increases strongly. Together, the two panels confirm strong impacts of AUC difference on probability of choosing the target image.

**Task 2.** In [T2], we predicted subjects would choose the higher-saliency scatterplot as showing clearer cluster structure. The probability of choosing this target option

should increase as differences in threshold bar lengths between the two response options grow. As described above, we modeled that probability that the subject would choose the target (higher-saliency) option in trial  $t$  using the bar lengths for the target (TLength) and comparison (lower-saliency; CLength) options and included random intercepts for each respondent ( $i$ ) and stimulus dataset ( $j$ ) to account for dependency across trials:

$$\begin{aligned} \text{ChooseTarget}_{ijt} &\sim \text{Bernoulli}(p_{ijt}) \\ \text{logit}(p_{ijt}) &= \beta_1 \times \text{TLength}_{ijt} + \beta_2 \times \text{CLength}_{ijt} + \\ &\quad \alpha + \gamma_i + \delta_j \\ \gamma_i &\sim \text{Normal}(\mu_\gamma, \sigma_\gamma) \\ \delta_j &\sim \text{Normal}(\mu_\delta, \sigma_\delta) \end{aligned} \quad (2)$$

**Analysis Software.** We fit models using the *glmmTMB* package [100, v. 1.1.1] in R [101, v. 4.1.0]. We computed and formatted model results using the *modelbased* [102] and *parameters* [103], [104] packages. We managed data using the *dplyr* [105] and *readr* [106] packages. We visualized model results using the *see* [107], *ggdist*, *ggplot2*, *ggtext*, and *patchwork* [108] packages.

## 6.5 Results

**Task 1.** Results for [T1] are shown in Table 2 and Fig. 12. Table 2 shows predicted probabilities of choosing the target option (Pr(CT)) based on the fitted model, with 95% confidence intervals, for combinations of target and comparison AUC values. Fig. 12 visualizes these results using a scatterplot showing individual trials with lines showing predicted probabilities of choosing the target option as a function of difference between the AUC values for the two scatterplot stimuli as options.

As shown, AUC differences strongly affected subject choice of which scatterplot was more similar to the reference plot. When the two stimuli options were equally SR (or DS) to the reference plot (i.e., when the AUC difference between the two plots' AUC values  $\approx 0$ ), participants tended to select both the target option and the comparison option at similar rates (Pr(CT) = .61 [.43, .77]). However, when the target plot was much more similar to the reference plot than the comparison plot (e.g., when the difference between their AUC values = 4), participants were much more likely to choose the target scatterplot (Pr(CT) = .95 [.99, .85]) when AUC difference = 3. This effect did not substantially vary across absolute levels of the target plot AUC (e.g., predicted probabilities for an AUC difference = 2 were similar regardless of whether the target plot was highly similar or only somewhat similar to the reference plot); this indicates that it is the difference in AUC values between the options that drives the change in subject choices. [H1] is validated.

**Task 2.** Results for [T2] are shown in Table 3 and Fig. 13. Table 3 shows predicted probabilities of choosing the target option (Pr(CT)) based on the fitted model, with 95% confidence intervals, for combinations of target and comparison threshold bar lengths. In the table, Low indicates short bar length (low saliency), Med indicates medium bar length (medium saliency), High indicates long bar length (high saliency). Fig. 13 visualizes these results using a

TABLE 3: Model predictions for [T2] show predicted probabilities of choosing the target option (Pr(CT)) based on the fitted model, with 95% CI, for combinations of target and comparison threshold bar lengths. Low indicates short bar length (low saliency), Med indicates medium bar length (medium saliency), High indicates long bar length (high saliency).

| Target length | Comp. length | Length diff. | Pr(CT) | SE   | 95% CI       |
|---------------|--------------|--------------|--------|------|--------------|
| Low [0.15]    | Low [0.15]   | 0.00         | 0.66   | 0.06 | [0.54, 0.76] |
| Med. [0.50]   | Med. [0.50]  | 0.00         | 0.62   | 0.06 | [0.51, 0.73] |
| High [1.50]   | High [1.50]  | 0.00         | 0.52   | 0.07 | [0.38, 0.66] |
| Med. [0.50]   | Low [0.15]   | 0.35         | 0.68   | 0.05 | [0.56, 0.77] |
| High [1.50]   | Med. [0.50]  | 1.00         | 0.68   | 0.05 | [0.57, 0.78] |
| High [1.50]   | Low [0.15]   | 1.35         | 0.73   | 0.05 | [0.62, 0.82] |

Pr(ChooseTarget) by Threshold Bar Length Difference

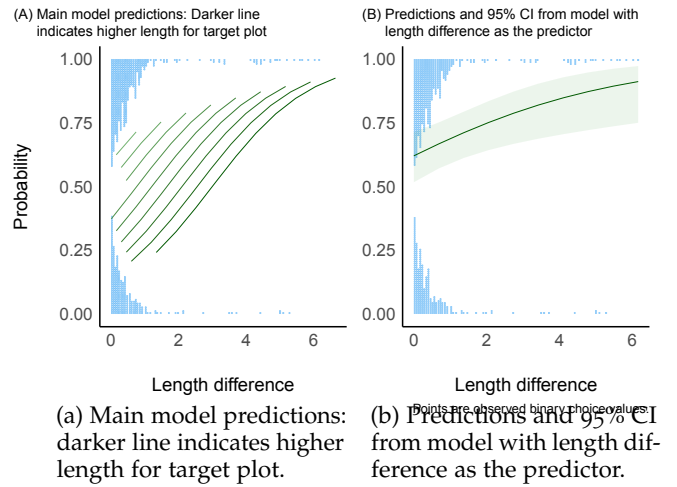


Fig. 13: User study analysis for [T2], Pr(ChooseTarget) by threshold bar length difference. The blue dot represents data points for each observation and lines show predicted probabilities of choosing the target option as a function of difference between the bar lengths for the two options. Panel (a) shows results for the main model. Each line reflects a combination of Target length and Comp. length as shown in Table 3. Darker lines indicate higher values of Target length. Each line shows that as the length difference increases, probability of choosing the target option increases strongly. However, the impact of length difference is most pronounced when the Target length is high (dark green lines), indicating that saliency contrast between images is most impactful when the target option has high saliency. Panel (b) shows results for an alternative model using length difference as the sole predictor. The line shows predicted probability of choosing the target option by levels of length difference, with 95% confidence band. This line shows that as the length difference increases, probability of choosing the target option increases. The slope of this line is somewhat shallow. Together, the two panels confirm strong impacts of length (saliency) difference on probability of choosing the target image, but also that such contrasts are most important when the target image has high saliency.

scatterplot showing individual trials with lines showing predicted probabilities of choosing the target option as a function of difference between the bar lengths for the two

options.

Bar length strongly affected participants' choice of which scatterplot was clearer. Participants were much more likely to choose the target scatterplot as having clearer cluster structure when there was a large difference in bar lengths between the two scatterplots. Participants were much more likely to choose the target scatterplot as having clearer cluster structure when there was a large difference in threshold bar lengths between the two scatterplots (e.g., .73 [.62, .82] for a length difference of 1.35 versus .52 [.38, .66] for a length difference of 0). Thus, saliency differences were a strong predictor of subject perceptions of cluster structure clarity. [H2] is validated. Bar length differences also strongly affected subject choice of which scatterplot showed clearer cluster structure.

## 7 CASE STUDY

To evaluate the quality of results and utility of the interface, we conducted a case study. We recruited ten graduate students from our departments who are researching visualization or taken a data visualization course but had not been previously exposed to our project or interface. To compare the utility of our interface and perform qualitative analysis, we also constructed a *manual optimization interface*, further referred to as M, which is different from our *user-guided optimization interface*, further referred to as A (see Sect. 4). The M interface featured sliders and option buttons for selecting scatterplot parameters, include sampling technique, number of points, point size, and opacity value.

We used the same datasets as in the user study (see Sect. 6). Each participant was asked to use the M and A interface for 3 datasets each to design a scatterplot. Dataset that was used for A and M were swapped between participants. The objective for each task (i.e., for each dataset) was to use the interface to select the factors (SA, SR, PS, and OP) that best highlight the clustering structure. The study was conducted in three parts: (1) instruction and training, (2) selecting optimal scatterplot, and (3) interview. The total time for each participant was approximately one hour. Each participants assigned eight datasets: four for M (one for training and three for tasks) and four for A (one for training and three for tasks).

The results of the study can be seen in Fig. 14, and the scatterplots from several subjects are shown in Fig. 15. We first investigate the number of interactions required, where an interaction is defined as selecting the values of the factors to select an optimal scatterplot. As one can observe in Fig. 14a, the manual optimization required a significantly higher number of interactions. In addition, in terms of time, we saw that the manual approach also tended to require more time from participants (see Fig. 14b). From conversations with participants, we hypothesize time is related to their confidence (less time, higher confidence) in the optimality of their choice, whereas the number of interactions is related to the usability of the interface (fewer interactions, higher usability).

In terms of quality, since all the participants had different datasets for the manual and automatic methods, we could not compare between subjects. However, Fig. 15 shows the output images for six datasets from two of the participants.

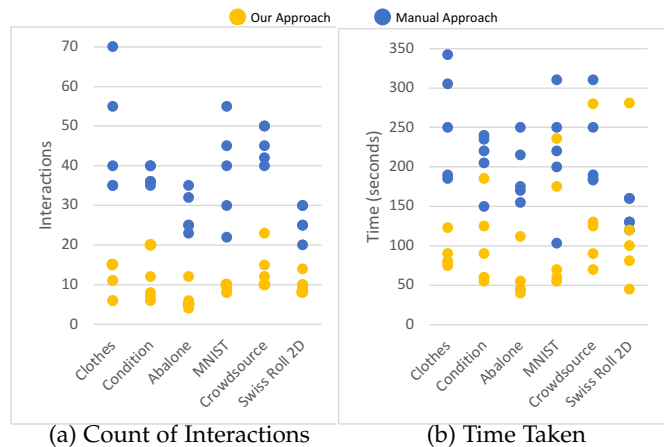


Fig. 14: Case study participants' performance in terms of the number of interactions and time for six different datasets. Each dot represents one participant's results (some are obscured by overlap). Each column represents one dataset.

Without the labeling (see Fig. 15), it is difficult to distinguish which are found using our interface (A) and which are found using the manual (M) approach. Second, each participant seemed to have their own preferred aesthetic, which they were able to produce in each interface.

## 8 DISCUSSION

The goal of our approach is to suggest an optimized visualization design to improve the effectiveness of the task performance, and it is important to understand how designers can use our models to reduce ambiguity in the data and thereby reduce the chance of misinterpretation, e.g., by having a visualization that is too sparse or over-saturated. Our approach uses a data-driven framework to compare and observe how cluster patterns change with a variety of density-influencing parameters of scatterplots, including point size and opacity visual encodings, as well as the subsampling algorithm and sampling rate. Our approach provides scatterplots ranked in order of their cluster saliency, where the saliency score (longest bar in the threshold plot) is a proxy for the clarity of the cluster structure in the scatterplot.

### 8.1 Saliency as a Proxy of Cluster Structure

The theoretical models of Sadahiro state that proximity, and number and concentration of points, and density change affect cluster perception [42]. Other experimental work has shown that the choice of visual factors which influence the visual density of scatterplot can have a significant effect on cluster identification [8]. The threshold plot is computed on the visual density estimate of the scatterplots and identifies how clusters visually merge together, fitting well with the known factors that influence clustering. With each bar of the threshold plot we measure the saliency of that number of clusters. In other words, how likely it is that a user will see that number of clusters. Therefore, by identifying the longest bar we capture the cluster structure most likely visible to the user.

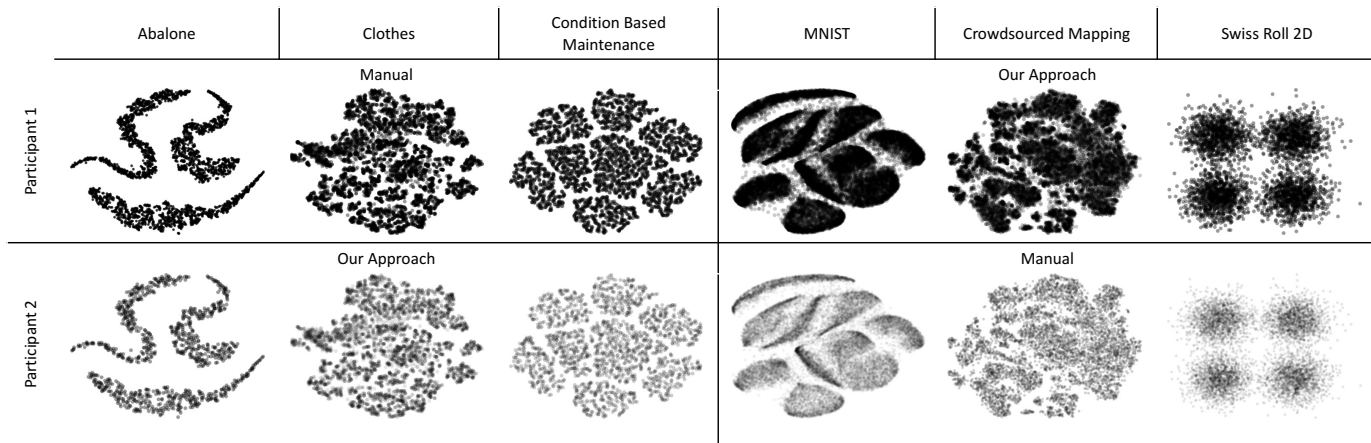


Fig. 15: Images created by two participants in the case study. Each image represents the optimized design chosen by the participants using a manual or our automatic approach.

## 8.2 Ambiguous Clustering Structure

Every dataset has an inherent properties that influence the visualization of the data. For example, data distribution plays a vital role in point concentrations concerning over-saturation or sparse distribution. Such properties often influence the visualization, leading to an ambiguous conclusion for even the optimal design choice (e.g., for Clothes and Crowdsourced Mapping datasets in Fig. 15). For these data, optimizing the design has negligible effects on clarifying the cluster structure which remains ambiguous for most parameter configurations. This problem also has a weak relationship with bin size at the time of threshold plot generation. Bins that are too large smooth the result, while bins too small create noise. This problem has been partially investigated previously [8] but deserves more attention in the future.

## 8.3 Comparisons to Existing Approach

*Visual Quality Measures (VQMs)* are ideally based on perceptual models rather than heuristics and computational approaches. The existing approaches, such as [17], [19], [27], apply measures that imitate how humans would score views (e.g., one or more clusters but not the specific count) based on the perceived patterns and can be used to accurately predict perceptual judgments. ClustMe used VQMs to model human judgments to rank scatterplots [27] which was further extended in [17]. These studies performed well in reproducing human judgments for quantifying cluster patterns as per points positions, but ignored the visual aspects (marker size, opacity, and visual density). Similarly, the *scagnostics* technique utilizes density property that identifies concentrations of points, which is directly influenced by the distribution of points [80] to investigate the patterns.

In contrast to these approach, our work proposes saliency score as a VQM that can be used to optimize design factors (data aspects and the visual encoding) and ranks the scatterplot designs on the cluster count that matches human understanding. It is important to note that the optimal design seemed to be both quantitative and qualitative. In other words, our saliency measure provided visualizations with clearer clustering structure, but each participant in our case study also seemed to have their own preferred aesthetic, which they were still able to produce with our interface.

## 8.4 Limitations

Our approach has some limitations. First, we have not considered some other factors that could influence performance in either model, e.g., chart size, screen resolutions, etc. We have also not extensively analyzed time variance between individuals' performance on the datasets and their sampling rate. We have not explored the histogram bin size to compute the density model, but the same is extensively discussed in [8]. A final limitation is that we have not considered the relationship of our approach to confidence [109], which is highly related to the nature of data [110].

## 9 CONCLUSIONS

Scatterplots are among the most powerful and most widely used techniques for visual data exploration of 2D data. Design choices in visualization, scatterplots in this case, such as the visual encodings or data aspects, can directly impact the quality of decision-making for low-level tasks, such as clustering.

We propose here a user-guided tool to optimize the design factors of scatterplot for salient cluster structure. By constructing frameworks, such as this one, that consider both the perceptions of the visual encodings and the task being performed enables maximizing the efficacy of the visualization. Our interactive tool leverages the application of the merge tree data structure to optimize the design decisions on sampling algorithms, sampling rate, symbol size, and opacity. We further validate our results with a user study, case studies, and demo interface that demonstrate guidelines that practitioners and designers can extend to other tasks on scatterplots.

Interface: <<http://scatter.projects.jadorno.com/>>

Data: <<https://osf.io/cxgq2/>>

## ACKNOWLEDGMENTS

The project is supported in part by the National Science Foundation IIS-1845204 and National Science Foundation CNS-2127309 to the Computing Research Association for the CIFellows program.

## REFERENCES

- [1] M. Friendly and D. Denis, "The early origins and development of the scatterplot," *J. of the History of the Behavioral Sciences*, 2005.
- [2] G. J. Quadri and P. Rosen, "A survey of perception-based visualization studies by task," *IEEE Transactions on Visualization and Computer Graphics*, 2021.
- [3] H. Nguyen, P. Rosen, and B. Wang, "Visual exploration of multiway dependencies in multivariate data," in *SIGGRAPH ASIA Symp. on Visualization*, 2016.
- [4] R. Rensink and G. Baldridge, "The perception of correlation in scatterplots," *Computer Graphics Forum*, 2010.
- [5] L. Harrison, F. Yang, S. Franconeri, and R. Chang, "Ranking visualizations of correlation using weber's law," *IEEE Trans. on Visualization and Comp. Graphics*, 2014.
- [6] M. Gleicher, M. Correll, C. Nothelfer, and S. Franconeri, "Perception of average value in multiclass scatterplots," *IEEE Trans. on Visualization and Comp. Graphics*, 2013.
- [7] A. Sarikaya, M. Gleicher, and D. Szafr, "Design factors for summary visualization in visual analytics," *Computer Graphics Forum*, 2018.
- [8] G. J. Quadri and P. Rosen, "Modeling the influence of visual density on cluster perception in scatterplots using topology," *IEEE Trans. on Visualization and Comp. Graphics*, 2020.
- [9] T. Munzner, *Visualization Analysis and Design*. CRC press, 2014.
- [10] R. Amar, J. Eagan, and J. Stasko, "Low-level components of analytic activity in information visualization," in *IEEE Symp. on Information Visualization (InfoVis)*, 2005.
- [11] Y. Kim and J. Heer, "Assessing effects of task and data distribution on the effectiveness of visual encodings," *Computer Graphics Forum*, 2018.
- [12] L. Micallef, G. Palmas, A. Oulasvirta, and T. Weinkauff, "Towards perceptual optimization of the visual design of scatterplots," *IEEE Trans. on Visualization and Comp. Graphics*, 2017.
- [13] D. Szafr, "Modeling color difference for visualization design," *IEEE Trans. on Visualization and Comp. Graphics*, 2018.
- [14] M. Sedlmair, A. Tatu, T. Munzner, and M. Tory, "A taxonomy of visual cluster separation factors," *Computer Graphics Forum*, 2012.
- [15] G. Ellis and A. Dix, "A taxonomy of clutter reduction for information visualisation," *IEEE Trans. on Visualization and Comp. Graphics*, 2007.
- [16] A. Sarikaya and M. Gleicher, "Scatterplots: Tasks, data, and designs," *IEEE Trans. on Visualization and Comp. Graphics*, 2018.
- [17] M. Aupetit, M. Sedlmair, M. M. Abbas, A. Baggag, and H. Bensmail, "Toward perception-based evaluation of clustering techniques for visual analytics," in *2019 IEEE Visualization Conference (VIS)*. IEEE, 2019, pp. 141–145.
- [18] J. Matute, A. Telea, and L. Linsen, "Skeleton-based scagnostics," *IEEE Trans. on Visualization and Comp. Graphics*, 2017.
- [19] M. Sedlmair and M. Aupetit, "Data-driven evaluation of visual quality measures," in *Computer Graphics Forum*, vol. 34, no. 3. Wiley Online Library, 2015, pp. 201–210.
- [20] Y. Ma, A. Tung, W. Wang, X. Gao, Z. Pan, and W. Chen, "Scatternet: A deep subjective similarity model for visual analysis of scatterplots," *IEEE Trans. on Visualization and Comp. Graphics*, 2018.
- [21] T. Dang and L. Wilkinson, "Transforming scagnostics to reveal hidden features," *IEEE Trans. on Visualization and Comp. Graphics*, 2014.
- [22] A. Pandey, J. Krause, C. Felix, J. Boy, and E. Bertini, "Towards understanding human similarity perception in the analysis of large sets of scatter plots," in *ACM SIGCHI Conference on Human Factors in Computing*, 2016.
- [23] R. Rensink, "On the prospects for a science of visualization," in *Handbook of human centric visualization*. Springer, 2014.
- [24] Y. Wang, X. Chen, T. Ge, C. Bao, M. Sedlmair, C.-W. Fu, O. Deussen, and B. Chen, "Optimizing color assignment for perception of class separability in multiclass scatterplots," *IEEE Trans. on Visualization and Comp. Graphics*, 2018.
- [25] K. Lu, M. Feng, X. Chen, M. Sedlmair, O. Deussen, D. Lischinski, Z. Cheng, and Y. Wang, "Palettailor: discriminable colorization for categorical data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 2, pp. 475–484, 2020.
- [26] Y. Wang, F. Han, L. Zhu, O. Deussen, and B. Chen, "Line graph or scatter plot? automatic selection of methods for visualizing trends in time series," *IEEE Trans. on Visualization and Comp. Graphics*, 2018.
- [27] M. Abbas, M. Aupetit, M. Sedlmair, and H. Bensmail, "Clustme: A visual quality measure for ranking monochrome scatterplots based on cluster patterns," *Computer Graphics Forum*, 2019.
- [28] B. Bederson, B. Shneiderman, and M. Wattenberg, "Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies," *ACM Transactions on Graphics*, 2002.
- [29] M. Derthick, M. Christel, A. Hauptmann, and H. Wactlar, "Constant density displays using diversity sampling," in *IEEE Symp. on Information Visualization (InfoVis)*, 2003.
- [30] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman, "Lifelines: visualizing personal histories," in *ACM SIGCHI Conference on Human Factors in Computing*, 1996.
- [31] A. Woodruff, J. Landay, and M. Stonebraker, "Constant density visualizations of non-uniform distributions of data," in *ACM Symp. on User interface software and technology*, 1998.
- [32] J. Matejka, F. Anderson, and G. Fitzmaurice, "Dynamic opacity optimization for scatter plots," in *ACM Conference on Human Factors in Computing Systems*, 2015.
- [33] E. Wegman and Q. Luo, "High dimensional clustering using parallel coordinates and the grand tour," in *Classification and Knowledge Organization*. Springer, 1997.
- [34] R. Kosara, S. Miksch, and H. Hauser, "Focus+ context taken literally," *IEEE Computer Graphics and Applications*, 2002.
- [35] J. Johansson, P. Ljung, M. Jern, and M. Cooper, "Revealing structure in visualizations of dense 2d and 3d parallel coordinates," *Information Visualization*, 2006.
- [36] J.-D. Fekete and C. Plaisant, "Interactive information visualization of a million items," in *IEEE Symp. on Information Visualization (InfoVis)*, 2002.
- [37] J. Cohen, "Eta-squared and partial eta-squared in communication science," *Human Communication Research*, 1973.
- [38] H. Chen, W. Chen, H. Mei, Z. Liu, K. Zhou, W. Chen, W. Gu, and K.-L. Ma, "Visual abstraction and exploration of multi-class scatterplots," *IEEE Trans. on Visualization and Comp. Graphics*, 2014.
- [39] D. Urribarri and S. Castro, "Prediction of data visibility in two-dimensional scatterplots," *Information Visualization*, 2017.
- [40] R. Hu, T. Sha, O. Van Kaick, O. Deussen, and H. Huang, "Data sampling in multi-view and multi-class scatterplots via set cover optimization," *IEEE Trans. on Visualization and Comp. Graphics*, 2019.
- [41] D. Szafr, S. Haroz, M. Gleicher, and S. Franconeri, "Four types of ensemble coding in data visualizations," *J. of Vision*, 2016.
- [42] Y. Sadahiro, "Cluster perception in the distribution of point objects," *Cartographica: The International Journal for Geographic Information and Geovisualization*, 1997.
- [43] S. Few and P. Edge, "Solutions to the problem of over-plotting in graphs," *Visual Business Intelligence Newsletter*, 2008.
- [44] M. Correll, M. Li, G. Kindlmann, and C. Scheidegger, "Looks good to me: Visualizations as sanity checks," *IEEE Trans. on Visualization and Comp. Graphics*, 2018.
- [45] G. Ellis and A. Dix, "Density control through random sampling: an architectural perspective," in *International Conference on Information Visualisation (IV)*, 2002.
- [46] A. Dix and G. Ellis, "By chance enhancing interaction with large data sets through statistical sampling," in *Working Conference on Advanced Visual Interfaces*, 2002.
- [47] E. Bertini and G. Santucci, "Improving 2d scatterplots effectiveness through sampling, displacement, and user perception," in *International Conference on Information Visualisation (IV)*, 2005.
- [48] J. Yuan, S. Xiang, J. Xia, L. Yu, and S. Liu, "Evaluation of sampling methods for scatterplots," *IEEE Trans. on Visualization and Comp. Graphics*, 2020.
- [49] D. Carr, R. Littlefield, W. Nicholson, and J. Littlefield, "Scatterplot matrix techniques for large n," *J. of the American Statistical Association*, 1987.
- [50] S. Bachthaler and D. Weiskopf, "Continuous scatterplots," *IEEE Trans. on Visualization and Comp. Graphics*, 2008.
- [51] D. Keim, M. Hao, U. Dayal, H. Janetzko, and P. Bak, "Generalized scatter plots," *Information Visualization*, 2010.
- [52] A. Mayorga and M. Gleicher, "Splatterplots: Overcoming overdraw in scatter plots," *IEEE Trans. on Visualization and Comp. Graphics*, 2013.
- [53] T. Trautner, F. Bolte, S. Stoppel, and S. Bruckner, "Sunspot plots: Model-based structure enhancement for dense scatter plots," *Computer Graphics Forum*, 2020.

- [54] J. Rojas, M. Kery, S. Rosenthal, and A. Dey, "Sampling techniques to improve big data exploration," in *IEEE Symp. on Large Data Analysis and Visualization (LDAV)*, 2017.
- [55] C. Palmer and C. Faloutsos, "Density biased sampling: An improved method for data mining and clustering," in *ACM SIGMOD International Conference on Management of Data*, 2000.
- [56] E. Bertini and G. Santucci, "Give chance a chance: modeling density to enhance scatter plot quality through random data sampling," *Information Visualization*, 2006.
- [57] —, "By chance is not enough: preserving relative density through nonuniform sampling," in *Conference on Information Visualisation*, 2004.
- [58] P. Joia, F. Petronetto, and L. G. Nonato, "Uncovering representative groups in multidimensional projections," *Computer Graphics Forum*, 2015.
- [59] Y. Zheng, J. Jests, J. Phillips, and F. Li, "Quality and efficiency for kernel density estimates in large data," in *ACM SIGMOD International Conference on Management of Data*, 2013.
- [60] X. Chen, T. Ge, J. Zhang, B. Chen, C.-W. Fu, O. Deussen, and Y. Wang, "A recursive subdivision technique for sampling multi-class scatterplots," *IEEE Trans. on Visualization and Comp. Graphics*, 2019.
- [61] J. Xia, F. Ye, W. Chen, Y. Wang, W. Chen, Y. Ma, and A. Tung, "Ldscanner: Exploratory analysis of low-dimensional structures in high-dimensional datasets," *IEEE Trans. on Visualization and Comp. Graphics*, 2017.
- [62] E. dos Santos Amorim, E. Brazil, J. Daniels, P. Joia, L. Nonato, and M. Sousa, "ilamp: Exploring high-dimensional spacing through backward multidimensional projection," in *IEEE Visual Analytics Science and Technology (VAST)*, 2012.
- [63] J. Poco, R. Etemadpour, F. V. Paulovich, T. Long, P. Rosenthal, M. d. Oliveira, L. Linsen, and R. Minghim, "A framework for exploring multidimensional data with 3d projections," *Computer Graphics Forum*, 2011.
- [64] B. Rieck and H. Leitte, "Persistent homology for the evaluation of dimensionality reduction schemes," *Computer Graphics Forum*, 2015.
- [65] S. Xiang, X. Ye, J. Xia, J. Wu, Y. Chen, and S. Liu, "Interactive correction of mislabeled training data," in *IEEE Visual Analytics Science and Technology (VAST)*, 2019.
- [66] S. Liu, J. Xiao, J. Liu, X. Wang, J. Wu, and J. Zhu, "Visual diagnosis of tree boosting methods," *IEEE Trans. on Visualization and Comp. Graphics*, 2017.
- [67] X. Zhao, W. Cui, Y. Wu, H. Zhang, H. Qu, and D. Zhang, "Oui! outlier interpretation on multi-dimensional data via visual analytics," *Computer Graphics Forum*, 2019.
- [68] S. Cheng, W. Xu, and K. Mueller, "Colormap nd: A data-driven approach and tool for mapping multivariate data to color," *IEEE Trans. on Visualization and Comp. Graphics*, 2018.
- [69] L.-Y. Wei, "Parallel poisson disk sampling," *ACM Transactions on Graphics*, 2008.
- [70] —, "Multi-class blue noise sampling," *ACM Transactions on Graphics*, 2010.
- [71] M. Berger, K. McDonough, and L. Seversky, "cite2vec: Citation-driven document exploration via word embeddings," *IEEE Trans. on Visualization and Comp. Graphics*, 2016.
- [72] M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM SIGMOD International Conference on Management of Data*, 2000.
- [73] J. Yellott, "Spectral consequences of photoreceptor sampling in the rhesus retina," *Science*, 1983.
- [74] D.-M. Yan, J.-W. Guo, B. Wang, X.-P. Zhang, and P. Wonka, "A survey of blue-noise sampling and its applications," *J. of Computer Science and Technology*, 2015.
- [75] C. Gramazio, K. Schloss, and D. Laidlaw, "The relation between visualization size, grouping, and user performance," *IEEE Trans. on Visualization and Comp. Graphics*, 2014.
- [76] W. Cleveland and R. McGill, "Graphical perception: Theory, experimentation, and application to the development of graphical methods," *J. of the American Statistical Association*, 1984.
- [77] B. Wong, "Points of view: gestalt principles," *Nature Methods*, 2010.
- [78] E. H. Cohen, M. Singh, and L. Maloney, "Perceptual segmentation and the perceived orientation of dot clusters: The role of robust statistics," *J. of Vision*, 2008.
- [79] G. Anobile, G. Cicchini, and D. Burr, "Number as a primary perceptual attribute: A review," *Perception*, 2016.
- [80] L. Wilkinson, A. Anand, and R. Grossman, "Graph-theoretic scagnostics," in *IEEE Symp. on Information Visualization (InfoVis)*, 2005.
- [81] R. Etemadpour and A. G. Forbes, "Density-based motion," *Information Visualization*, 2017.
- [82] R. Veras and C. Collins, "Saliency deficit and motion outlier detection in animated scatterplots," in *ACM SIGCHI Conference on Human Factors in Computing*, 2019.
- [83] H. Chen, S. Engle, A. Joshi, E. D. Ragan, B. Yuksel, and L. Harrison, "Using animation to alleviate overdraw in multiclass scatterplot matrices," in *ACM SIGCHI Conference on Human Factors in Computing*, 2018.
- [84] J. Heer, N. Kong, and M. Agrawala, "Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations," in *ACM SIGCHI Conference on Human Factors in Computing*, 2009.
- [85] D. Chung, D. Archambault, R. Borgo, D. Edwards, R. Laramée, and M. Chen, "How ordered is it? on the perceptual orderability of visual channels," *Computer Graphics Forum*, 2016.
- [86] P. Rosen, J. Tu, and L. A. Piegl, "A hybrid solution to parallel calculation of augmented join trees of scalar fields in any dimension," *Computer-Aided Design and Applications*, 2018.
- [87] A. Zomorodian and G. Carlsson, "Computing persistent homology," *Discrete & Computational Geometry*, vol. 33, no. 2, pp. 249–274, 2005.
- [88] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.
- [89] A. Coraddu, L. Oneto, A. Ghio, S. Savio, D. Anguita, and M. Figari, "Machine learning approaches for improving condition-based maintenance of naval propulsion plants," *Proceedings of the Institution of Mechanical Engineers, Part M: J. of Engineering for the Maritime Environment*, 2016.
- [90] B. Johnson and K. Iizuka, "Integrating openstreetmap crowd-sourced data and landsat time-series imagery for rapid land use/land cover (lulc) mapping: Case study of the laguna de bay area of the philippines," *Applied Geography*, 2016.
- [91] R. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Physical Review E*, 2001.
- [92] M. Sedlmair, T. Munzner, and M. Tory, "Empirical guidance on scatterplot and dimension reduction technique choices," *IEEE Trans. on Visualization and Comp. Graphics*, 2013.
- [93] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [94] R. Kohavi *et al.*, "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid," in *Kdd*, vol. 96, 1996, pp. 202–207.
- [95] H. Kaya, P. Tüfekci, and E. Uzun, "Predicting co and no x emissions from gas turbines: novel data and a benchmark pems," *Turkish journal of electrical engineering & computer sciences*, vol. 27, no. 6, pp. 4783–4796, 2019.
- [96] I.-C. Yeh and C.-h. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert systems with applications*, vol. 36, no. 2, pp. 2473–2480, 2009.
- [97] B. Strack, J. P. DeShazo, C. Gennings, J. L. Olmo, S. Ventura, K. J. Cios, and J. N. Clore, "Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records," *BioMed Research International*, 2014.
- [98] StatCounter, "Desktop screen resolution stats worldwide," <http://gs.statcounter.com/screen-resolution-stats/desktop/worldwide>, 2019.
- [99] C. G. Akcora, Y. Li, Y. R. Gel, and M. Kantarcioglu, "Bitcoinheist: Topological data analysis for ransomware detection on the bitcoin blockchain," *arXiv preprint*, 2019.
- [100] M. Brooks, K. Kristensen, K. Van Benthem, A. Magnusson, C. Berg, A. Nielsen, H. Skaug, M. Machler, and B. Bolker, "glmmTMB balances speed and flexibility among packages for zero-inflated generalized linear mixed modeling," *The R Journal*, 2017.
- [101] R Core Team, "R: A language and environment for statistical computing," [Online]. Available: <http://www.r-project.org/>
- [102] D. Makowski, D. Lüdtke, M. Ben-Shachar, and I. Patil, "modelbased: Estimation of model-based predictions, contrasts and means."

- [103] D. Lüdecke, M. Ben-Shachar, I. Patil, and D. Makowski, "Extracting, computing and exploring the parameters of statistical models using R," *J. of Open Source Software*, 2020.
- [104] D. Lüdecke, D. Makowski, M. Ben-Shachar, I. Patil, S. Højsgaard, and B. Wiernik, "parameters: Processing of model parameters." [Online]. Available: <https://CRAN.R-project.org/package=parameters>
- [105] H. Wickham, R. François, L. Henry, and K. Müller, "dplyr: A grammar of data manipulation." [Online]. Available: <https://dplyr.tidyverse.org/>
- [106] H. Wickham and J. Hester, "readr: Read rectangular text data." [Online]. Available: <https://readr.tidyverse.org/>
- [107] D. Lüdecke, I. Patil, M. Ben-Shachar, B. Wiernik, P. Waggoner, and D. Makowski, "see: An R package for visualizing statistical models," manuscript submitted for publication.
- [108] T. L. Pedersen, "patchwork: The composer of plots." [Online]. Available: <https://CRAN.R-project.org/package=patchwork>
- [109] D. Sacha, H. Senaratne, B. C. Kwon, G. Ellis, and D. A. Keim, "The role of uncertainty, awareness, and trust in visual analytics," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 240–249, 2015.
- [110] R. Etemadpour, R. Motta, J. G. de Souza Paiva, R. Minghim, M. C. F. De Oliveira, and L. Linsen, "Perception-based evaluation of projection methods for multidimensional data visualization," *IEEE Trans. on Visualization and Comp. Graphics*, 2014.



**Paul Rosen** is an Associate Professor at the University of Utah. He received his Ph.D. from the Computer Science Department of Purdue University. His research interests include applying geometry- and topology-based approaches to problems in information visualization. Along with his collaborators, he has received awards for best paper at PacificVis 2016, IVAPP 2016, PacificVis 2014, and SIBGRAPI 2013. Dr. Rosen received a National Science Foundation CAREER Award in 2019.



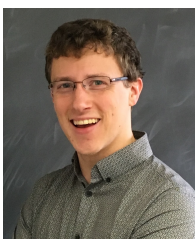
**Ghulam Jilani Quadri** is a CIFellow postdoc at the University of North Carolina at Chapel Hill in the Department of Computer Science. He received his PhD degree from University of South Florida. Before joining the University of South Florida, Ghulam worked for Infosys Limited as System Engineer in Pune. Ghulam and team participated in IEEE VIS 2017 VAST Challenge and awarded Honorable mention. Ghulam received 2021 Computing Innovation Fellow award. His research interests are Information Visualization

and HCI.



**Jennifer Adorno Nieves** is a PhD Candidate of Computer Science and Engineering at the University of South Florida. Her dissertation work revolves around machine learning systems on public transportation for improving predicted arrival times as part of the Location Aware Information Systems Lab and in collaboration with the Center for Urban and Transportation Research. Additionally, she works with mobile technologies to develop cost effective approaches that can be used for diagnosis and remote monitoring of

patients within the healthcare field. Generally, Jennifer's areas of interest include: Ubiquitous Sensing, Health and Accessibility. Over the summers, she serves as a mentor on the Research Experience for Undergraduates program hosted at her home institution.



**Brenton M. Wiernik** is an Affiliate Professor at the University of South Florida in the Department of Psychology. Dr. Wiernik is currently an independent researcher and Research Scientist at Meta, Demography and Survey Science. The current research was conducted while he was employed at the University of South Florida. He received his Ph.D. from the University of Minnesota. His recent research interests include psychometrics, latent variable modeling, and personality assessment. He is a developer of numerous R packages and

Senior Editor for Organizational Behavior at Collabora: Psychology.