October 2021

Constructing Frameworks for Task-Optimized Visualizations

Ghulam Jilani Abdul Rahim Quadri

University of South Florida

Follow this and additional works at: https://digitalcommons.usf.edu/etd

Part of the Computer Sciences Commons, Engineering Commons, and the Library and Information Science Commons

Scholar Commons Citation
https://digitalcommons.usf.edu/etd/9213

This Dissertation is brought to you for free and open access by the USF Graduate Theses and Dissertations at Digital Commons @ University of South Florida. It has been accepted for inclusion in USF Tampa Graduate Theses and Dissertations by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact digitalcommons@usf.edu.
Constructing Frameworks for Task-Optimized Visualizations

by

Ghulam Jilani Abdul Rahim Quadri

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science and Engineering
College of Engineering
University of South Florida

Major Professor: Paul Rosen, Ph.D.
Shaun Canavan, Ph.D.
John Licato, Ph.D.
Mahshid Naeini, Ph.D.
Brenton Wiernik, Ph.D.

Date of Approval:
October 11, 2021

Keywords: Perception, Visual Design, Optimization, Clustering, Line Smoothing

Copyright © 2021, Ghulam Jilani Abdul Rahim Quadri
Dedication

To my grandfather for his vision for education. To every person who was there for me.
Acknowledgments

The work in this dissertation would not have been possible without the mentorship of my amazing advisor. I owe tremendous gratitude to my advisor Dr. Paul Andrew Rosen. His guidance has been valuable through my Ph.D. journey. He has always been patient, helpful, understanding, and insightful from the very beginning. I thank him for all the time, energy, and teaching me so much over these last five years. His ability to show you the way without telling you where to go at times is incredible.

I want to thank Dr. Shaun Canavan, Dr. John Licato, Dr. Mahshid Naeini, and Dr. Brenton Wiernik for serving on my committee and Dr. Alon Friedman for chairing the defense. I am delighted to have had the opportunity to work with everyone on my committee. Again, I would like to thank Dr. Brenton Wiernik for his feedback on chapter 3 and guidance and collaboration on chapter 6.

I would also like to acknowledge the impact of our research group, Graphics and Visualization Lab, throughout my time here at USF. Thank you, Dr. Zach Beasley, Dr. Junyi Tu, and Bhavana Doppalapudi. I want to thank the staff of the CSE main office, especially Gabriela Franco, Laura Owczarek, and Jessica Pruitt, for their help and the entire department for their warmth and kindness.

I would also like to thank my friends in Computer Science and Engineering department, from whom I learned so much— Dr. Sulav Malla, Dr. Anwesh Tuladhar, Dr. Jean-Baptiste Subils, Dr. Rahul Paul, Dr. Saeed Alhamari, Dr. Ghada Zamzami, Jennifer Adorno, and Steven Diaz. They always provided at least one pair of eyes whenever needed, for the many unforgettable discussions
and the occasional much-needed distraction. I want to express my gratitude to Jennifer Adorno for her help and collaboration in chapter 6. Thank you again to Sulav and Anwesh for boosting my technical skills and providing your valuable guidance during the challenges of my Ph.D. days.

I also owe thanks to my uncle Aziz Makki, aunt Tajbi Makki, Afiya, Atif, Kausar, Marie, John, Nadia, Ellie, Aaisha, Arsh, and Awan. I want to express my sincere gratitude to my mentor and supporters— Dr. Abbas Khatkhtay, Dr. Nasreen Fazalbhoy, Yasmeen Barelvi, Shahbaz Siqqiqui, and Dr. Ali Qureshi. They continually believed in me even in my lowest and held my hand to show me the right path.

A big thank you to all my friends at Tampa— Karan, Jaleel, Sameer, Mudassir, Kartik, Komal, Arshad, Manaswini, Pratishruti, Prejith (to name a few) for all the time spent together from trips to hangouts to having delicious food. Gratitude to Arshad, Manaswini, Pratishruti for being there during the COVID-19 pandemic. A huge thank you to my friend in India who never made me feel away from home— Rachita; thank you, Rach, for being a content editor for all my papers and presentations.

I am incredibly indebted to God, without whose will and blessings, Ph.D. would have been a far-fetched dream.

This work was supported in part by the U.S. National Science Foundation under Grant No. IIS-1845204. Any opinions, findings, and conclusions or recommendations expressed in this project are those of author(s) and do not necessarily reflect the views of the National Science Foundation. The dissertation writing was also funded by the Dissertation Writing Fellowship from the College of Graduate Studies.
# Table of Contents

List of Tables ................................................................................ viii

List of Figures................................................................................ xi

Abstract ..................................................................................... xii

Chapter 1: Introduction ...................................................................... 1
  1.1 Motivation....................................................................... 3
  1.1.1 Problem Organization .................................................. 5
  1.2 Contributions.................................................................... 5
  1.3 Outline of Dissertation........................................................... 10

Chapter 2: Background ...................................................................... 12
  2.1 Perception Fundamentals ........................................................ 12
  2.1.1 Graphical Perception ................................................... 12
  2.1.2 Psychophysical Effects.................................................. 15
    2.1.2.1 Psychophysics................................................. 15
    2.1.2.2 Weber’s Law.................................................. 16
    2.1.2.3 Just-Noticeable Difference (JND).............................. 16
  2.1.3 Bias and Effect......................................................... 17
    2.1.3.1 Perceptual Bias ............................................... 17
    2.1.3.2 Cognitive Bias ................................................ 18
  2.2 Visualization for Task Effectiveness .............................................. 20
    2.2.1 Analytical Tasks in Scatterplots ........................................ 21
    2.2.2 Analytical Tasks in Line Graphs........................................ 22
      2.2.2.1 Task Efficacy.................................................. 22
      2.2.2.2 Decision-Making .............................................. 24
  2.3 Optimizing Visualization Design ................................................. 25

Chapter 3: A Survey of Perception-Based Visualization Studies by Task ......... 27
  3.1 Introduction ..................................................................... 27
    3.1.1 Systematic Survey Literature........................................... 30
      3.1.1.1 Examples for Inclusion ........................................ 31
      3.1.1.2 Examples for Exclusion........................................ 31
    3.2 Structure of the Taxonomy................................................... 32
      3.2.1 Low-Level Tasks.................................................. 32
      3.2.2 Taxonomy on Visual Encoding.......................................... 33
      3.2.3 Visualization........................................................... 35
      3.2.4 Survey.................................................................. 36
      3.2.5 Discussion.............................................................. 37
3.3 Retrieve Value

3.3.1 Visual Encoding
- 3.3.1.1 Spatial Position, Shape, and Size
- 3.3.1.2 Color Hue
- 3.3.1.3 Color Intensity

3.3.2 Visualizations
- 3.3.2.1 Scatterplot
- 3.3.2.2 Bar Chart
- 3.3.2.3 Parallel Coordinates
- 3.3.2.4 Text
- 3.3.2.5 Map

3.4 Filter

3.4.1 Visual Encoding
- 3.4.1.1 Spatial Position, Shape, and Size
- 3.4.1.2 Color Hue

3.4.2 Visualizations
- 3.4.2.1 Parallel Coordinates
- 3.4.2.2 Map
- 3.4.2.3 Text

3.5 Compute Derived Value

3.5.1 Visual Encoding
- 3.5.1.1 Spatial Position, Shape, and Size
- 3.5.1.2 Color Hue and Intensity
- 3.5.1.3 Other

3.5.2 Visualization
- 3.5.2.1 Bar Chart
- 3.5.2.2 Pie Chart
- 3.5.2.3 Maps

3.6 Find Extremum

3.6.1 Visual Encoding

3.6.2 Visualization
- 3.6.2.1 Bar Chart
- 3.6.2.2 Line Chart
- 3.6.2.3 Network and Map

3.7 Determine Range

3.7.1 Visualization
- 3.7.1.1 Bar Chart
- 3.7.1.2 Line Chart

3.8 Sort

3.8.1 Visualization
- 3.8.1.1 Bar Chart
- 3.8.1.2 Parallel Coordinates
- 3.8.1.3 Text
- 3.8.1.4 Other

3.9 Find Anomalies

3.9.1 Visual Encoding
- 3.9.1.1 Spatial Size
- 3.9.1.2 Color Hue
- 3.9.1.3 Color Intensity
4.4 Case Study ...................................................................... 90
  4.4.1 Graphical Perception ................................................... 91
  4.4.2 Example Replication Studies ............................................ 92
    4.4.2.1 Re-evaluate an Experiment’s Objective ....................... 92
    4.4.2.2 Expand an Experiment’s Objective ................................. 93
    4.4.2.3 Specialize an Experiment’s Objective .............................. 94
4.5 Discussion ....................................................................... 95
  4.5.1 You Can’t Publish (Strict) Replication Studies ................. 95
  4.5.2 How to Publish Replication Studies Anyways .................. 96

Chapter 5: Modeling Cluster Perception in Scatterplots ............ 99
5.1 Introduction ..................................................................... 99
  5.1.1 Clustering in Scatterplots ............................................ 100
    5.1.1.1 Taxonomies of Clustering Factors ............................... 101
    5.1.1.2 Clustering in Non-Scatterplot Contexts ....................... 101
  5.1.2 Factor Selection on Scatterplots ................................... 102
    5.1.2.1 Point Distribution ......................................................... 102
    5.1.2.2 Number of Data Points .................................................. 102
    5.1.2.3 Size of Data Points ....................................................... 103
    5.1.2.4 Opacity of Data Points .................................................. 103
5.2 Study Methodology .......................................................... 104
  5.2.1 Factors ....................................................................... 104
  5.2.2 Experiments Setup ...................................................... 105
  5.2.3 Data Generation .......................................................... 105
5.3 Preliminary Experiment ....................................................... 106
  5.3.1 Properties and Data Generation .................................... 107
  5.3.2 Study Procedure .......................................................... 108
  5.3.3 Analysis and Result ....................................................... 109
5.4 Topology-based Modeling of Clustering ............................... 110
  5.4.1 Distance-based Model .................................................. 111
  5.4.2 Density-based Model .................................................... 113
  5.4.3 Persistence Threshold Plot ............................................. 115
5.5 Main Experiment .............................................................. 117
  5.5.1 Properties and Data Generation .................................... 117
  5.5.2 Study Procedure .......................................................... 118
    5.5.2.1 Suitability of Studying Point Size Using AMT ............... 119
  5.5.3 Analysis Methodology ..................................................... 119
  5.5.4 Results ....................................................................... 120
    5.5.4.1 Model Accuracy ......................................................... 120
    5.5.4.2 Factor Effect Analysis Without a Model ....................... 121
    5.5.4.3 Distance-based Model Factor Analysis ......................... 123
    5.5.4.4 Density-based Model .................................................... 124
    5.5.4.5 Histogram Resolution .................................................. 124
    5.5.4.6 Number of Points Model (T\textsubscript{N}/D\textsubscript{N}) .............. 125
    5.5.4.7 Point Size Model (T\textsubscript{P}/D\textsubscript{P}) .................................. 126
    5.5.4.8 Interaction Model (T\textsubscript{N}\times\text{P}/D\textsubscript{N}\times\text{P}) ...................... 126
    5.5.4.9 Post-Test Questionnaire .............................................. 126
5.6 Model Usage ................................................................. 127
  5.6.1 Controlling Design Factors .......................................... 128
8.2 Limitations ........................................................................................................ 219
  8.2.1 Design Optimization in Scatterplots and Line Charts ...................... 219
  8.2.2 Task-optimized Framework .............................................................. 220

8.3 Future Work ................................................................................................... 221
  8.3.1 Consider Other Visualizations ....................................................... 222
  8.3.2 Consider Other Visual Encoding .................................................... 222
  8.3.3 Multi-Visualization Optimization System Deployment ................... 222
  8.3.4 Improving the Effectiveness of User-Guided Optimization Tool ....... 223
  8.3.5 Diverse Participatory Recruitment for User Study Evaluation ........ 223

References .............................................................................................................. 248

Appendix A: Copyright Permissions for Chapter 3, 4 and 7 ............................. 249

Appendix B: Institutional Review Board Authorization ..................................... 252
List of Tables

Table 3.1 The number of surveyed papers by source.......................................... 31
Table 3.2 Low-level taxonomy with icon representation and description based on [10]........ 34
Table 3.3 Visual encoding taxonomy used in the survey......................................... 35
Table 3.4 Number of studies reviewed per task, visual encoding, and visualization......... 79
Table 4.1 Selected publications replicating studies of Cleveland and McGill [58]. .......... 90
Table 5.1 Distance-based model Repeated-Measures ANOVA results. ........................ 123
Table 5.2 Density-based model Repeated-Measures ANOVA results. ........................... 124
Table 5.3 Summary of the main effects found in the study based on $\eta^2$.................... 128
Table 6.1 Sampling algorithm implemented in this study and their application............... 145
Table 6.2 Model parameters for task 1............................................................... 173
Table 6.3 Model predictions for task 1............................................................... 173
Table 6.4 Model parameters for task 2............................................................... 174
Table 6.5 Model predictions for task 2............................................................... 175
Table 7.1 Summary of smoothing algorithms analyzed. ........................................ 189
Table 7.2 Matrix of tasks and metrics .................................................................... 197
Table 7.3 Grades for the efficacy of different smoothing methods.............................. 214
List of Figures

Figure 1.1  Eight representation of similar dataset with different cluster count .......... 4
Figure 1.2  Chicago homicide rates data for weekly and monthly samples .............. 4
Figure 2.1  Reproduction of visual encodings in [58] ...................................... 13
Figure 2.2  Reproduction of the Mackinlay visual encoding rankings [177] .......... 13
Figure 2.3  Example of attraction effect in selecting hypothetical election candidates ... 18
Figure 3.1  Publications per year in this survey considering perception since 2000 ....... 28
Figure 3.2  Number of publications utilizing crowdsourcing experiments in recent years .... 29
Figure 3.3  Reproduction of visual encodings in [58] ...................................... 33
Figure 3.4  Visualization list category for taxonomy ........................................ 36
Figure 3.5  A study evaluated a pairwise relation between visualization types across tasks .... 39
Figure 3.6  Examples of text visualizations generated by visual encodings and layouts [100] .... 45
Figure 3.7  Perceived color difference varies inversely with size, and elongated marks [280] .... 66
Figure 3.8  Ranking the effectiveness of visualizations for representing correlation [121] .... 70
Figure 3.9  Ranking the effectiveness of visualizations for representing correlation [142] .... 71
Figure 4.1  Reproduction of the Cleveland and McGill’s graphical encoding channels [58] .... 91
Figure 5.1  Selected factors for experiment and user study ................................ 104
Figure 5.2  Illustration of the data generation for user study ............................. 106
Figure 5.3  Example stimuli with the same cluster centers but varying factors .......... 107
Figure 5.4  The histogram of user responses against frequency ............................ 109
Figure 5.5 Distance-based model representation ............................................. 112
Figure 5.6 Density-based model representation .................................................. 114
Figure 5.7 The persistence threshold plots ..................................................... 116
Figure 5.8 Plots of the threshold and differential ............................................. 121
Figure 5.9 Histograms for user response differential against frequency ............... 122
Figure 5.10 Normalized persistence threshold for the density-based model for the data .... 125
Figure 5.11 Density-based model mean and 95% CI of user response on factors ........... 127
Figure 5.12 Example of overplotted scatterplot stimuli ..................................... 130
Figure 5.13 Demonstration of density-based model ............................................ 131
Figure 6.1 Illustration of design choice optimization model ..................................... 144
Figure 6.2 Demonstration of user-guided interactive model ................................... 156
Figure 6.3 Demonstration of threshold plot generation using the visual density .......... 157
Figure 6.4 The threshold plot for cluster structure saliency .................................. 158
Figure 6.5 Cluster structure ranking using threshold plot ...................................... 160
Figure 6.6 The threshold plots showing the similar patterns are perceptually similar ........ 160
Figure 6.7 Binning for saliency and area and under curve distribution .................... 161
Figure 6.8 Data sampling time analysis for 14 sampling algorithms ......................... 165
Figure 6.9 Time analysis for projection and feature extraction .............................. 166
Figure 6.10 Model scalability time analysis ...................................................... 167
Figure 6.11 User study analysis for task 1 ....................................................... 174
Figure 6.12 User study analysis for task 2 ....................................................... 175
Figure 6.13 Case study participants performance for time and number of interaction .... 177
Figure 7.1 Representation of line smoothing framework ...................................... 183
Figure 7.2 Illustration of local and global smoothing methods for line charts .......................... 187
Figure 7.3 Illustration of the 5 effectiveness measurement used in developed approach ........ 192
Figure 7.4 Example stock price line chart for demonstrating task queries.......................... 196
Figure 7.5 Low-level task presentation on smoothed line charts ........................................ 198
Figure 7.6 Low-level task presentation on smoothed line charts ........................................ 199
Figure 7.7 Low-level task presentation on smoothed line charts ........................................ 200
Figure 7.8 Low-level task presentation on smoothed line charts ........................................ 201
Figure 7.9 Entropy plot for the EEG Channel 10 (500 samples) dataset ............................ 203
Figure 7.10 Examples of datasets used to analyze smoothing technique ............................ 204
Figure 7.11 Rank plot of the $L^1$-norm for all EEG (500 samples) datasets ....................... 205
Figure 7.12 Average ranking for Retrieve Value and Determine Range tasks ..................... 209
Figure 7.13 Average ranking for the Compute Derived Value task .................................... 210
Figure 7.14 Average ranking for the Find Extrema and Anomalies tasks .......................... 211
Figure 7.15 Average ranking for the Distribution and Cluster: Trends tasks ...................... 212
Figure 7.16 Average ranking for the Sort and Cluster: Points tasks ................................. 213
Abstract

Visualization is crucial in today’s data-driven world to augment and enhance human understanding and decision-making. Effective visualizations must support accuracy in visual task-performance and expressive data communication. Effective visualization design depends on the visual channels used, chart types, or visual tasks. However, design choices and visual judgment are co-related, and effectiveness is not one-dimensional, leading to a significant need to understand the intersection of these factors to create optimized visualizations. Hence, constructing frameworks that consider both design decisions and the task being performed enables optimizing visualization design to maximize efficacy. This dissertation describes experiments, techniques, and user studies to model user perception for visualization design optimization and data transformation for low-level visual tasks. To begin with, I identify the limitations through a taxonomized state-of-the-art survey on perception-based visualization studies focusing on how visualization effectiveness is task-dependent. With a specific focus on the scatterplot, I developed perceptual models for cluster perception and design optimization. In addition to design guidelines from the first experiments, I employ the findings to show design choices based on the visual density of the scatterplot could influence the user’s judgment on visual tasks. Further, I address the challenge of assessing line chart smoothing effectiveness for a range of analytical tasks. Finally, I elaborate on utilizing the framework to provide less ambiguous data presentations, leading to better quality and higher confidence in decision-making.
Chapter 1: Introduction

“The greatest value of a picture is when it forces us to notice what we never expected to see.” — J.W. Tukey, Exploratory Data Analysis, 1977.

Data are regarded as the “fuel of the future” [84], and through visualization tools, we communicate the information in the data. Visualization has become a crucial tool in today’s data-driven world at communicating underlying structure and patterns in information to augment and enhance human understanding and decision-making. Almost every sub-field of science and analytics utilizes data visualization. It supports dynamic exploration, reveals patterns, and characterizes distributions in small-to-large, simple-to-complex, univariate-to-bivariate-to-multivariate, and different types of datasets—temporal and spatial, time-series, and intensity quantitative name a few. Effective visualization must provide a clear and unbiased understanding of information in the given data to support users’ accuracy in visual task-performance and expressive data communication. However, the current information visualization tools have a limited hold on the optimization challenge of providing an effective visualization design.

The way data are visualized, in terms of both visual channels, e.g., colors, symbol-types, and sizes [58, 280, 303], and data aspects, e.g., number of data points or sampling rate (for subsampled data), affects the user performance on analysis tasks. Both visual design and chart types can influence and drastically change the conclusions people draw using data. For example, analysis of blood-flow visualization is influenced by choice of visual channels (colors or mark types), affecting
Effective visualizations improve understanding of data by leveraging visual perception, e.g., the size of marks in scatterplots better represents quantitative data, while color can express categorical data. In addition, the effectiveness of the visualization varies with the tasks being performed on it, e.g., when searching for clusters vs. outliers, the opacity impacts the visibility of data.

Visualization design aims directly at improving the user’s data interpretation and accuracy on visual task performance. The influence on judgment with visual design is evaluated using methods such as human understanding and perception [58, 285]. Key findings in the graphical perception studies [58] generate design guidelines by measuring accuracy with different visual encodings of data on quantitative-value tasks. Most of these guidelines focus on either specific low-level tasks or visualization design optimization itself. Design optimization using visual encoding [147, 280] and data aspect improves the user performance on the low-level visual tasks [10], data interpretation, and communication. However, the current state of art design optimization is per visualization e.g., bar chart [282], pie charts [265], scatterplots [300]. Findings around visualization effectiveness are generally studied on singular aspects leading to a significant need to understand the intersection of these factors to create optimized visualization.

Research on design recommendations focuses on effective visualization design that depends on the visual channels used, chart types, or visual tasks, but independently. Visualization effectiveness in terms of their visual design choice and task helps the analysts make sense of the datasets at hand. However, we learned that design choices and visual judgment are co-related, and effectiveness is not one-dimensional [221], leading to a significant need to understand the intersection of these factors to
create optimized visualizations. Hence, constructing frameworks that consider both design decisions with human perception and the task being performed enables optimizing visualization design to maximize efficacy.

1.1 Motivation

A design framework at these intersections of visual encoding and low-level tasks fills the gap between guidelines and application by providing a task-optimized visualization design. An optimal strategy is for better quality and higher confidence in decision-making that gives designers objective guidance. This work is different from previous work in developing a framework that investigates the task effectiveness of visualization design in an ensemble compared to individual visual design efficacy. I import the perceptual design guidelines on visual encoding and data aspects in visualization design. Further, I introduce it to a framework model intending to present an optimal design that suggests the parameters for the factors, provides a perceptual model based on the factors, and analytical ranking for data transformation in context to visual tasks.

At a high level, some visualizations can provide better efficacy or higher accuracy for a given visual task and dataset in hand. Studies in the last decade demonstrated that not all visualizations are fit for every task [244], or some visualization is suitable for a given task, e.g., a correlation on scatterplot [121], or visual encoding choice performance varies based on visualization performance on tasks [147]. For example, as shown in Figure 1.1, the cluster count varies with factors, whether it is point size, opacity, number of points, or even data features of distribution. Providing an optimal solution in terms of visual design choice of scatterplot for clustering task would help visualization designer optimize for effective data communication. Similarly, to improve the task performance,
we optimize the visualization representation, e.g., line smoothing for noisy data (Figure 1.2). But, choosing a data transformation or visualization representation further heeds evaluation of best choice, such as 1) which smoothing technique is better for a given task, or 2) which design of horizon-graph provides highest estimation accuracy [124].

Figure 1.1: Eight representation of similar dataset with different cluster count. A cluster is grouping of similar object or data points. In this case, all eight representation of similar dataset with varying point size, number of data points and point distribution size gives different perception of cluster count.

Figure 1.2: Chicago homicide rates data for weekly and monthly samples. Data contains weekly (969 samples) and monthly (222 samples) counts of the number of homicides in the city from January 2001 through July 2019. Data is provided by the City of Chicago [52]. The current visualization is not optimal for a visual task required to identify the maximum value because of the apparent noise.
1.1.1 Problem Organization

This work investigates visual encoding, task, and visualization to develop task-optimized guidelines for effective data communication, including 1) Task-Oriented Visualization that develop task-optimized frameworks to generate visualizations that enable rigorous scientific exploration for a given analysis task. The design-optimized frameworks suggest a set of visual encodings and related parameters (e.g., symbol-opacity, symbol-size), including those to guide data selection and data aspect (e.g., number of data points or sampling rate), and data transformation method (e.g., line smoothing). 2) Data Communication that conducts evaluation using user study by recruiting participants from crowdsourcing mediums. The results are quantified to improve the data communication on widely used visualizations.

As my goal is to improve the effectiveness of the visualization design, to begin with, it is essential to understand the influence of different visual encoding, data aspects, and data transformation on design and task performance. In addition, designers can use the models to reduce ambiguity in the data and thereby reduce the chance of misinterpretation, e.g., by having a too sparse or over-saturated visualization for clustering tasks in a scatterplot. Furthermore, data transformation can lead to a better visual design that improves the user performance accuracy on low-level tasks, such as smoothing in line graphs for noisy data.

1.2 Contributions

This dissertation addresses the need for a task-optimized framework and overcomes the demand for the intersectional study of task and visual design in visualization in three distinctive parts.
1. In the first part of this dissertation, I considered the progress over the last two decades on perception-based studies and the need for a comprehensive guide to contextualize their results. In particular, I focus on how perception is used to evaluate the effectiveness of visualizations to help understand and apply the principles of perception of visualization designs through a task-optimized approach. The findings from the survey aided me in building the design optimization foundation for the following two parts of my dissertation.

2. Identifying clusters is an essential low-level visual analytics task [10], as well as in data analysis in general [202]. Still, clustering is an ill-posed problem, with the “correct result” subject to the algorithm’s constraints or individual performing the analysis. Scatterplots are used for various visual analytics tasks, including cluster identification. The visual encodings used on a scatterplot play a deciding role in the visual separation of clusters. For visualization designers, optimizing the visual encodings is crucial to maximizing data clarity, and it requires accurately modeling human perception of cluster separation, which remains challenging. In this second part of the dissertation, I address the requirement of accurately modeling human perception of cluster separation. In particular, I investigate how the visual encodings and data aspects influence the cluster perception in the scatterplot and apply the findings to optimize the visual encodings to maximize cluster saliency.

3. Line charts are commonly used to visualize a series of data samples. When the number of samples is large, or the data are noisy, smoothing can be applied to make the signal more apparent. This process optimizes the visual design of line charts. However, there are a wide variety of smoothing techniques available, and the effectiveness of each depends upon both nature of the data and the visual analytics task at hand. To date, the visualization community
lacks a summary work for analyzing and classifying the various smoothing methods available.

In the third part of the dissertation, I establish a framework based on eight measures of the line smoothing effectiveness tied to eight low-level visual analytics tasks. I then analyze 12 methods coming from four commonly used classes of line chart smoothing—rank filters, convolutional filters, frequency domain filters, and subsampling, and investigate the ranking of smoothing techniques based on how optimal they are for a given task.

My approach applies the theory from perception to design the experiments and analyze them at a fundamental level and it uses algorithms and data structure from topological data analysis to develop perceptual models and analytical framework. I then apply findings in a loop-centered approach to validate the solution for one part of the dissertation and a technique-based analytical strategy to evaluate and suggest the optimal choice for another part. Findings comprise several experimental, fundamental-level, analytical, model-based, and theoretical contributions. The contributions of this dissertation are as follows:

- **Survey of Perception-Based Visualization**
  - *Taxonomy of Perception-Based Visualization Studies:* I review perception-focused visualization studies since 1980 and summarize their research developments into a practical taxonomy on low-level tasks, further breaking techniques down by visual encoding and visualization type. My important observations from this survey are—I witnessed the continuously evolving nature of perceptual research, with the majority focusing on visual task judgment; low-level task effectiveness varies with the dataset at hand, the visualization used, and specific design variations with the visualization. The discussion around the findings also explores the need to study less frequently studied techniques. While this
survey is not exhaustive, it provides a taxonomy for considering the task, visual encoding, and visualization in terms of graphical perception.

– Replication in Visualization: One of the critical challenges with many graphical perception studies is their limited scope and reproducibility, partly caused by the difficulty of constructing human studies. To scratch the replication surface in visualization research, I defined three methods—re-evaluation, expansion, and specialization—for embedding a replication study into a novel published work. I provide a non-exhaustive case study on replications of Cleveland and McGill’s seminal work on graphical perceptions. As it turns out, numerous replication studies have been carried out based on that work, which has both confirmed prior findings and shined new light on the understanding of human perception. I discuss how trying to publish a true replication study should be avoided while providing suggestions for how others can still use replication studies as a vehicle to producing visualization research publications.

- Scatterplot Design Optimization

– Design Guidelines of Visual Factors: I present a multi-stage user study focusing on four factors—distribution size of clusters, number of points, size of points, and opacity of points—that influence cluster identification in scatterplots, and the analysis demonstrates that these factors play an important role in the number of clusters perceived. While some variables, such as distribution size, are difficult to control in a visualization, designers can use these models and findings as a guideline to balance the design factors that they do have control over—the number of points shown, data point size, or opacity—to optimize the saliency of the clusters in a visualization.
- **Topology-based Perceptual Models:** I have constructed two models, a distance-based model and a density-based model, using the merge tree data structure from Topological Data Analysis. The analysis demonstrates that the above factors play an essential role in the number of clusters perceived. It verifies that the distance-based and density-based models can reasonably estimate the number of clusters a user observes. Finally, I demonstrate how these models can be used to optimize visual encodings on real-world data.

- **Design Optimization Tool:** I applied design guidelines of visual factors on topology-based models and developed the user-guided optimization tool. I utilized scatterplot visual design factors under the designer's control as input to the tool. The data-driven interface provides an optimal scatterplot design based on the range of factors chosen. The results are validated on user study against user understanding, and findings suggest optimization model can be used as a proxy for cluster structure saliency. The proposed optimal design solution focuses on the clustering task in a scatterplot and discusses how the designer and practitioner can extend it to other low-level tasks.

- **Line-Smoothing Framework**

  - **Analytical Framework for Line Smoothing Techniques:** Designers turn to smoothing to reduce the visual clutter in line graphs. However, there are many techniques available, and while the results they produce may look similar, each preserves different properties of the data. To preserve some properties of the input data, each smoothing technique must also lose information, which can have a negative impact on the utility of the resulting data. To further complicate matters, the importance of the lost information can be influenced by both the data being used and the visual analytics tasks being performed.
Here, I present a framework of 12 different smoothing techniques under four categories, measuring effectiveness and preserving properties.

– *Ranking of Line Smoothing Techniques*: There is a wide variety of smoothing techniques available, and the effectiveness of each depends upon both nature of the data and the visual analytics task at hand. To date, the visualization community lacks a summary work for analyzing and classifying the various smoothing methods available. In this work, I establish a ranking framework based on eight measures of the line smoothing effectiveness tied to eight low-level visual analytics tasks. I then analyze 12 methods coming from four commonly used classes of line chart smoothing—rank filters, convolutional filters, frequency domain filters, and subsampling. Visualization designers can use this framework and results to select a smoothing technique, evaluate their data to determine the method that is specifically most effective, or understand how much error is introduced as they increase the level of smoothing used in their visualizations.

1.3 **Outline of Dissertation**

The rest of the dissertation is organized as follows. Chapter 2 provides an overview of graphical perception, fundamentals of perception, and design optimization in visualization.

Chapter 3 presents the state-of-the-art survey for perception-based studies and provides a systematic and comprehensive review using a taxonomy on low-level tasks. The findings of the surveyed literature are presented using a taxonomy on eleven different low-level tasks, four categories of visual encodings, and ten visualization types. The *web interface* provides an interactive list of
papers. Chapter 4 defines three methods for embedding a replication study into a novel publication work.

Chapter 5 models the influence of visual encoding and data aspects in scatterplot design on cluster perception with an application of *merge tree*. The experiments investigate the effect of factors on visual density leading to misinterpretation of cluster count identification. The model proposes and validates that it can reasonably estimate the number of clusters a user observes. Chapter 6 architects, the design optimization of a scatterplot by applying the visual parameters affects finding using a merge tree-based models. The new algorithm suggests an optimal scatterplot based on a user-guided factor selection tool.

Chapter 7 demonstrates the importance of line smoothing in noisy data and presents an analytical framework that solves the challenge of choosing the optimal smoothing technique for a given task. Based on the metrics introduced, all methods are evaluated and ranked based on their performance.

Chapter 8 summarizes and quantifies the potential need for task-based optimized visualization in this dissertation and provides a discussion around the limitation and future research direction.
Chapter 2: Background

2.1 Perception Fundamentals

I provide a brief introduction to some fundamental concepts of perception discussed in this dissertation. The presented information is concise, and readers are encouraged to refer to the related articles for more in depth information.

2.1.1 Graphical Perception

Data visualizations encode information using a variety of visual encodings, including position, length, angle, area, volume, shading, direction, curvature, and color (see Figure 2.1) [25]. The term graphical perception—the visual encoding of information encoded on graphs, was introduced by Cleveland & McGill in 1984 [58]. They further ranked visual encodings in terms of perceptual precision by measuring the perceptual magnitude of judgments to determine their accuracy, with some encoding types (e.g., area and color saturation) being less perceptually precise than others (e.g., position). The terms graphical encoding, visual channel, visual encodings, and visual properties are often interchanged, but generally, they mean the same thing. Throughout the remainder of this dissertation, I refer to them as visual encodings.

Understanding the role of perception in the choice of visual encodings is critical to visualization designers. Visualization has incorporated knowledge and theories from perception into visual design [122]. Most guidelines for effective visualization design come from graphical perception—the
study of how well viewers can interpret different encodings of data. Mackinlay produced the first comprehensive ranking of visual encodings by data type, as shown in Figure 2.2 [177]. The ranking has been further validated and elucidated through numerous follow-up studies, e.g., [71, 111, 123, 124, 191, 226, 246, 256, 280, 282].

In the recent last two decades, visualization research has seen an increase in the utilization of human perception in visual design effectiveness studies. Human perception has had a wide-ranging

![Figure 2.1: Reproduction of visual encodings types in Cleveland and McGill’s work [58].](image)

![Figure 2.2: Reproduction of the Mackinlay visual encoding rankings [177].](image)
impact on effective visualizations' design. The studies demonstrated that perception helps designers understand what the visual system can and cannot do with different visual information. The perception literature and application of perception fundamentals in visualization studies explored the means to provide effective design and helps us understand limitations in visual processing, design cases, user studies. Numerous fields of science have studied perception, including perceptual psychology, visualization, and human-computer interaction. Much of our understanding of perception in visualization is rooted in an early work that ranked the order of visual encodings based on their effectiveness for visual judgment [58]. Chapter 3 provides a taxonomized list of perception-based studies that measured the efficacy of visual design using *graphical perception* when the user performs low-level tasks on visualizations.

Graphical perception is an anchor of visualization research [303] with the potential to improve the efficiency and effectiveness of the visual design. We have observed the vast influence of this paper in the last two decades of information visualization research. According to Google Scholar, this work has been cited more than 2030 times as of September 1, 2021, and aspects of their study have been replicated many times. Cleveland and McGill’s seminal work on Graphical perception acts as commencing practice to focus on the visualization effectiveness and originated many oncoming perceptual studies with key findings and results.

Heer et al. explained how various designs for time series data improve how viewers find and compare particular values [124]. Haroz and Whitney assessed how grouping like items could improve viewers’ abilities to find target values [118]. Szafir’s color difference model indicated people’s abilities to perceive color differences varies significantly with marks types, e.g., colors are more discriminable on bars and lines than on points [280]. Kim et al. measure the impact of task and data distribution
on the effectiveness rankings of visual encodings [147]. These experiments provide grounded guidance
and methodologies for reasoning about the effectiveness of visualization design approaches. However,
they focus on a fixed task or set of tasks or involving small groups of visualization.

This dissertation characterizes visualization effectiveness differently than previous approaches
at an intersection of visual design and low-level tasks making the effectiveness two-dimensional.
This work is different from previous work in developing a framework that investigates the task
effectiveness of visualization design in an ensemble compared to individual visual design efficacy.

Studies in perception contribute remarkable evidence as to how optimized visualization
designs can be generated. Perception research focuses on how the visual system interprets low-level
abstract features, such as color and position, and how the accuracy and performance vary with their
design choice. Chapter 3 surveys several perception-based visualization studies that measured the
efficacy of the visual design.

2.1.2 Psychophysical Effects

2.1.2.1 Psychophysics

Psychophysics is a set of methods relating sensations to the characteristics of a stimulus. It
is used to quantitatively investigate relationships between physical stimuli and the sensation of the
perception they produce [105, 106]. The transition of psychology from a philosophical to a scientific
discipline was facilitated when G.T. Fechner introduced techniques to measure mental events. The
attempt to measure sensations through the use of Fechner’s procedures was termed psychophysics
and primarily investigated the relationships between sensations in the psychological domain and
stimuli in the physical domain. Central to psychophysics is the concept of a sensory threshold, that
measurement can have a differential and an absolute sensitivity. The *absolute threshold* or *stimulus threshold* is defined as the smallest amount of stimulus energy necessary to produce a sensation. The *differential threshold* was defined as the amount of change in a stimulus required to produce a just noticeable difference (JND) in the sensation.

2.1.2.2 Weber’s Law

Weber-Fechner’s law, relates to psychophysics and is used to determine the relationship between the perceived change in a stimulus and the actual change, which has been used to modeling how humans perceive certain features in a visualization. The law states that the change in a stimulus that will be just noticeable is a constant ratio of the original stimulus [306, 308]. As an example of Weber’s law, consider the act of lifting a 5 kg weight. Adding a small amount of weight, say 0.1 kg, may not make the weight *feel* any heavier. With further additions of weight, the difference will eventually be noticeable. Weber’s law is the ratio of change in the stimulus \( \Delta I = 0.1 \) kg to the stimulus magnitude \( I = 5 \) kg, which is 0.02. Weber’s law has been shown to hold for weight discrimination, visual discrimination, and tone discrimination [307].

2.1.2.3 Just-Noticeable Difference (JND)

JND also known as the difference threshold, is the minimum level of stimulation that a person can detect, usually 50% of the time, though other ratios can be used [64]. For example, one is asked to hold two objects of different weights—the JND would be the minimum weight difference between the two that one could sense half of the time. JND is used as a component for the perceptual studies, which are required to determine how much a given stimulus must be regulated in order for a human to detect a change reliably. The relation between JND and the stimulus can be represented using
Weber’s law [306, 308] as follows: 
\[ dP = k \frac{dI}{I}, \]
where, \( dP \) is the differential change in perception; \( k \) is the Weber fraction; \( dI \) is the differential change in stimulus, and \( I \) is the actual intensity of the stimulus. With the given \( I \) and Weber fraction, JND corresponds to the minimum change of stimulus will produce a noticeable difference in the perception. The application of JNDs with psychophysics evaluation is useful for measuring human judgments in the effectiveness of visual encodings for different tasks or design improvements.

2.1.3 Bias and Effect

In the process of evaluating visualizations, the effect of bias, whether cognitive or perceptual, is critical to understanding and evaluating experimental results. The primary types of bias observed and studied in visualization are perceptual biases and cognitive biases.

2.1.3.1 Perceptual Bias

Perceptual biases are systematic errors that occur at the perceptual level in perceiving visualization and/or related tasks. Some types of perceptual biases studied include clustering illusions and priming biases [292]. For example, the clustering illusion is where people underestimate the variance seen in patterns in a small set of random data [187, 257]. Priming relates to associative memory theories where a concept/effect is quickly activated after a similar concept/effect has been activated.

Perceptual bias has been studied in several regards. Change blindness refers to the inability of humans to recognize large visual changes between images. An optimization-based method introduced and evaluated an approach to generate “spot-the-difference” alternatives [175]. Perceptual biases have been studied in virtual reality platforms, e.g., in the perception of size and stiffness of virtual
objects [323] and how photorealism negatively affects our perception of virtual characters [339]. Irregularities in data can cause bias, influencing a user’s response to the conclusion of analysis [67, 269]. One study found that perceptual biases influence a user’s awareness of uncertainties, further influencing the user’s trust building [241]. Finally, researchers have investigated how bias and perception intersect to create deceptive views. Pandey et al. developed a method to quantify and compare the exaggeration caused by misleading representations [207].

2.1.3.2 Cognitive Bias

Cognitive bias research has grown considerably at both the cognitive science level and specifically for visual analytic and decision-making tools. Cognitive biases differ from perceptual biases in that they persist even if the information has been correctly processed at a perceptual level. Broadly, collective works on cognitive biases can be found in [90]. The following are examples of cognitive bias topics that have been evaluated in visualization.

- Attraction effect is a type of cognitive bias where the presence of irrelevant alternatives influences the choice between two options. When the user’s choice between two options is influenced by the presence of an irrelevant (dominated) third alternative. A good example, coming from [277], involves selecting between two candidates based on a rating of their education and crime control plans (see Figure 2.3). Bob has a solid education plan, while

<table>
<thead>
<tr>
<th>education</th>
<th>Bob</th>
<th>Alice</th>
<th>(Eve)</th>
</tr>
</thead>
<tbody>
<tr>
<td>crime control</td>
<td>★★★★★</td>
<td>★★</td>
<td>★★★★</td>
</tr>
<tr>
<td></td>
<td>★★★★</td>
<td>★★★★</td>
<td>★★★★</td>
</tr>
<tr>
<td></td>
<td>★★★</td>
<td>★★★★</td>
<td>★★★★</td>
</tr>
<tr>
<td></td>
<td>★★★</td>
<td>★★★★</td>
<td>★★★★</td>
</tr>
</tbody>
</table>

Figure 2.3: Example of attraction effect in selecting hypothetical election candidates.
Alice’s strategy for crime control is excellent. The choice is difficult if we consider both the
criteria. A third candidate, Eve, called the “decoy,” is focused more on crime control than
education, though not as good as Alice. The introduction of Eve as alternative biases our
preference towards Alice. Awareness of the attraction effect is important, as it may introduce
skew into decision-making tasks, as evidenced by Dimara et al.’s study on scatterplots [74].

- Anchoring effect is a type of cognitive bias, where a stimulus might influence human judgment
  at the perceptual level of the decision-making process. Anchoring effects and ordering effects
describe how the order in which information is presented can affect the perceived size of an
effect, with subjects across a wide range of domains tending to assign more rhetorical weight
to evidence that comes near the beginning of a sequence.

- Priming is another form of stimulus that can influence human judgment. The priming effect is
  a phenomenon in which an alternative perceptual stimulus influences human responses [184].
  Priming effects are seen more frequently than anchoring effects in separability judgment.

- Emotions, in psychology, are defined by two dimensions: valence, positive or negative feelings,
  and arousal, the intensity of the feelings. Affective priming is the technique of inducing emotion,
  also known as affect, in a user to study the impact on cognitive tasks. It involves manipulating
  valence and/or arousal via emotional stimuli. Harrison et al.’s contribution to affective
  priming suggested that it can influence accuracy in graphical perception tasks [119, 120]. The
  crowdsourced study indicated that affective priming significantly influenced visual judgment,
  while positive priming improved accuracy. The cognitive biases of both anchoring and priming
  suggest that the decision-making process not only depends on what the visual features currently
  look like but also on the previous frame of reference.
2.2 Visualization for Task Effectiveness

Effectiveness criteria determine whether a graphical language exploits the capabilities of the output medium and the human visual system [177]. The visual representation is “not just about making pretty pictures and graphs” instead designing a visualization that could be easily interpreted, data can be communicated, and the user’s performance on task wins on the metric of correctness, accuracy, and principle of graphical integrity. However, the effectiveness of visualizations types significantly varies across tasks, suggesting that visualization design would benefit from considering context-dependent effectiveness [244]. For this dissertation, I select two commonly used visualization for experimentation, scatterplot and line graph.

Scatterplots are commonly used to reveal several types of relationships between quantitative variables [104]. Numerous perceptual studies have evaluated the effectiveness of scatterplots in low-level tasks that include assessing trends [77, 196], measuring correlation [30, 121, 230], and average and relative mean judgments [109]. Clustering, in particular, is an aggregate-level task [166, 188, 247] that has been utilized in a variety of applications, e.g., weather forecasting, text analysis, and large-scale data analysis [165, 281, 304, 319].

Line charts, which date back to William Playfair [215], are commonly used for visualizing time-series and continuous data. Borkin et al. found that line charts are the second most frequently used visualization type, only behind bar charts, in scientific publications, news media, government, and world organizations materials [34].
2.2.1 Analytical Tasks in Scatterplots

Several prior perceptual studies have demonstrated the effect of visual encodings on analysis tasks [58, 111, 280]. A variety of factors influence group or separation perception [320], including color, size, shape [256], orientation [59], texture [12], opacity [186], density [317], motion and animation [49, 94, 296], chart size [124], and others. Other studies have demonstrated a perceptual effect in scatterplots when changing factors in the data, including data distribution types, number of points, the proximity of concentrations of points, data point opacity, and relative density [56, 67, 109, 111, 147, 242, 280]. Overdraw in scatterplots, in particular, has been addressed with a variety of techniques, e.g., splatterplot [183], recursive sampling [50], set cover optimization [128], feature-preserving visual abstraction [48], or by applying various clutter reduction techniques [92], e.g., sampling [76, 89, 91, 311] or changing opacity [180].

Clustering plays an important role in exploring and understanding many types of data [247, 248]. A design factor survey defined clustering as a high-level data characterization—the ability to identify groups of similar items [248]. Amar et al. presented a set of tasks for visual analytics that defined clusters as having “similar attribute values in a given set of data” [10].

Several works have considered how to model the perception of clusters. For example, a recent study that used eye-tracking to analyze user perception in cluster identification, highlighted the role of Gestalt principles, especially proximity and closure [97]. Matute et al. provided a method to quantify and represent scatterplots through skeleton-based descriptors that measured scatterplot similarity [181]. However, their approach does not consider visual encodings in the evaluation. ScatterNet, a deep learning model, captures perceptual similarities between scatterplots to emulate human clustering decisions but lacks explainability in the choices [176]. The scagnostics technique
focused on identifying the patterns in scatterplots, including clusters [69]. However, a study by Pandey et al. showed that they do not reliably reproduce human judgments [206]. Recently, ClustMe used visual quality measures to model human judgments to rank scatterplots [1]. ClustMe performed well in reproducing human decisions for clustering patterns. In contrast, I am studying the extent to which various factors influence the perception of clusters and building explainable models of how humans perceive cluster separation using the merge tree data structure.

2.2.2 Analytical Tasks in Line Graphs

I discuss prior work in the context of analytical tasks performed using line charts, decision-making with line charts, and line chart smoothing.

2.2.2.1 Task Efficacy

Line charts, which are traditionally used to visualize time-series and continuous 1D data [154, 287, 335], have been studied in the context of a variety of low-level visual analytics tasks [10]. A recent multi-chart experimental study found that line charts are significantly more accurate than other charts for the tasks of correlation and, to a lesser extent, finding extrema, characterizing distributions, and filtering [244]. Even so, line charts are used for a wider variety of visual analytics tasks.

- Comparison Tasks: Early work on horizon graphs [243] investigated their effectiveness as compared to line charts in a comparison task [124]. The work identified space-accuracy trade-off that could be used to optimize perception between the two. Another study compared line charts to horizon graphs and colorfields for similarity assessment [110]. The study showed
that deformations in the data are perceived differently depending on the visualization, and, in particular, line charts are more sensitive to changes in amplitude than position.

- **Statistical Tasks**: Line charts are used in many forms of statistical analysis [198]. Perception-based experiments that measure user’s judgments concluded that line charts have low-to-medium precision on estimating correlation [121, 142]. More generally, when considering aggregation tasks, it has been shown that line charts are effective at finding minima and maxima and determining value range while falling short on determining the average, spread, and outliers in the data [5]. More specifically, when calculating averages, colorfields have been shown to outperform line charts [65].

- **Trend Assessment**: Another common task attended to with line charts is trend assessment. It has been shown that line charts are, in general, better at trend assessment than scatterplots and bar charts, particularly for nonlinear trends [31]. However, when outliers are introduced into the data, the trends in their estimates begin to diverge from standard regression models [66]. Furthermore, when data are noisy, trends in the data are easier to identify using scatterplots [301].

- **Visual Encoding, Layout, and Interaction**: The visual encodings, layout, and interaction with line charts can have an impact on their efficacy. For example, color is an important visual encoding. For line charts, it has been shown that color difference varies inversely with thickness [280]. In other words, to be effective, a light-colored line must be thicker than dark-colored ones. Concerning layout, the efficacy of line charts is subject to the choice of aspect ratio, which can be automatically optimized for a chart [302]. Javed et al. evaluated the effectiveness of line charts with small multiples, horizon graphs, stacked graphs, and
braided graphs for comparison, slope, and discrimination tasks [134]. The results showed that techniques with separate charts performed better for data with large visual spans, while shared-space techniques were better for short spans. A more recent study showed that overlaid line charts perform better than small multiples in comparison tasks [201]. Finally, adding user interaction to line charts can enhance the user experience without a loss to efficacy [2].

2.2.2.2 Decision-Making

Line charts have been studied in several decision-making scenarios as well. In time-sensitive application settings, the ability to accurately interpret a line chart “at a glance” is crucial. Recently, Pixel Approximate Entropy (PAE) was used as a metric for the perceptual complexity of line charts, and it was shown that increased chart PAE correlates with reduced judgment accuracy [240].

- Missing Data: Another decision-making challenge in visualization is when data are missing; one needs to impute the missing data into the visualization [283]. An early study that looked at the problem of missing data in line charts on trend and comparison tasks found that even with missing data, user performance was high [83]. One way to address missing data is to use additional visual channels, e.g., color, empty points, or error bars, which have been shown to improve analysis performance and confidence of users on average and trend finding tasks [269].

- Distortion: Another concern for decision-making is distortion in the visualization, such as an inverted axis or a distorted aspect ratio. Such situations can demonstrate a reversal of messaging, which can lead viewers to draw false inferences and judgments [207]. Another example is when missing data are misleadingly inserted into a visualization, e.g., assigned
arbitrary values, user performance can go down significantly [83]. To address this weakness, multi-view systems have been proposed to assist in time-series data quality checking [13].

2.3 Optimizing Visualization Design

Optimizing visualization design is a perennial topic in the Human-Computer interaction and visualization community. Optimizing a design is to choose visual encodings that maximize the saliency of the visualization and data clarity. It also improves estimation and judgment accuracy in task performance. Here, I discuss how optimization of visual design has been rigorously studied under different banners.

- Design Recommendations: Design choices and recommendations form a critical element of effective visualization. In his book [303], Colin Ware talked about human perception in the context of information visualization design. He aimed at broadly summarizing the design implications of research in perception and suggested explicit design guidelines. In their survey, Healey and Enns focused more specifically on the role that attention and visual memory play in the perception of visualizations [122]. The work highlights how what users see impacts the viewer’s accuracy in information judgment. Finally, VisGuides is a web-based forum, which was established to collect practical knowledge of visualization guidelines and feedback on designs [73].

- Frameworks: In addition to general guidelines, several researchers have focused on developing robust frameworks for optimizing design. Rensink’s framework for reasoning about perceptions of visualization designs suggests using techniques from vision science [229]. The extended-vision theory asserts that a viewer and visualization system is a single system, whereas the
optimal-reduction thesis postulates an optimal visualization. The work focuses on a few of the fundamental questions, e.g.: *What is the best way to measure how a given visualization works?* Or, *could we determine if its design is optimal?* Recent work by Elliott et al. introduced a design space of experimental methods for empirically investigating the perceptual processes involved in viewing data visualizations to inform visualization design guidelines [88]. The chapter provides shared design space and lexicon for facilitating empirical visualization research. Researchers can use this design space to create innovative studies and progress scientific understanding of design choices and evaluations in visualization. In contrast, I provide an overview of perception-based studies by surveying papers that have evaluated the effectiveness of various visualization under a variety of tasks.

- Evaluation: When optimizing a design, measurements of effectiveness are critical to understanding their impact. Studies often rely upon subject evaluations of visualizations. One important innovation is the introduction of crowdsourcing environments which are faster, less costly, and provide more diverse subject pools than lab-based studies (Figure 3.2). Borgo et al. provided a detailed review of the use of crowdsourcing for evaluation in visualization research [33]. In addition to subject evaluations, there are many methods and metrics for quantitatively evaluating the effectiveness of a visualization. Behrisch et al. [22] gave an extensive audit of the state-of-art in quality metrics for various visualization techniques, along with details on a variety of implementation possibilities. The papers I discuss throughout this survey use a combination of both subject evaluations and quantitative measures to formulate their conclusions.
Chapter 3: A Survey of Perception-Based Visualization Studies by Task

“Visualization is where information theory meets psychology. — Min Chen, EuroVis, 2021.”

3.1 Introduction

Visualization provides valuable assistance in data analysis and decision-making tasks. The human perceptual and cognitive systems are essential in the process of visualization, influencing visual analysis activities, e.g., data exploration, data gathering, and data manipulation. As an example, data exploration requires forming high-level analysis goals, planning actions, and evaluating results effectively, all of which are cognitive activities. Before higher-level cognitive processes analyze data, visualization passes through the human perceptual system impacting the visualization’s utility [285]. The design of the visualization should make it as easy and unambiguous as possible to understand the data. Ultimately, a better understanding of human perception aids visualization design in both a quantitative and qualitative manner [303] \(^1\).

Numerous fields of science have studied perception, including perceptual psychology, visualization, and human-computer interaction. Much of our understanding of perception in visualization is rooted in an early work that ranked the order of visual encodings based on their effectiveness for visual judgment [58]. The findings were pivotal in nature and a milestone in research, demon-

\(^1\)This chapter was published in IEEE Transactions on Visualization and Computer Graphics (Early Access) 2021, DOI: 10.1109/TVCG.2021.3098240. Permission included in Appendix B
strating that the application and understanding of perception lead to guidelines for an effective and expressive visual design. Furthermore, numerous works on improving and evaluating visualization’s effectiveness have utilized the knowledge of attention, psychophysics, stimulus, judgment estimation, and perceptual laws to confirm the importance of perception throughout the process of generating a visualization [77, 115, 118, 122, 228, 280], and we have observed a notable increase in the interest in perception-based visualization studies over recent decades (see Figure 3.1).

In this chapter, I review perception-focused visualization studies since 1980 and summarize their research developments. The focus is primarily on information visualization—areas such as scientific visualization, 3D perception, etc., have been surveyed elsewhere, e.g., [218]. Within this context, I create a practical taxonomy of prior studies based upon Amar et al.’s low-level task taxonomy [10], further breaking techniques down by visual encoding and visualization type. In particular, I focus on how perception is used to evaluate the quality of visualizations, to help readers understand and apply the principles of perception to their visualization designs through a task-optimized approach.

![Figure 3.1: Publications per year in this survey considering perception since 2000.](image-url)
I focus this survey on studies that measured the efficacy of the visual design using *graphical perception* when the user is performing various low-level tasks. Furthermore, since perception and cognition are not entirely separable, I discuss some perceptual effects on cognitive performance, e.g., completion time, accuracy, and error rate. Unless otherwise noted, throughout the remainder of this survey, I will use the term *perception* to refer to *graphical perception*.

One of the key challenges with many graphical perception studies is their limited scope and reproducibility [219], caused in part by the difficulty of constructing human studies. Human studies are often resource-limited by the types and sizes of data, variety of visualizations or visual encodings, tasks being performed, and the size and diversity of subject pools, to name a few. This all puts transferability of results on tenuous footing. There is mounting evidence of the perceptual efficacy for many tasks and visualization types, but this survey provides a window into open research questions in many other situations. One of this survey’s goals is collecting and organizing findings in such a way that the reader can make a judgment of the applicability of results to their context.

![Number of publications utilizing crowdsourcing experiments in recent years.](image)

Figure 3.2: Number of publications utilizing crowdsourcing experiments in recent years. Reproduction with permission [33].
3.1.1 Systematic Survey Literature

Perception, being a critical part of visualization, has a wide range of applications and is included throughout different visualization-related journals and conferences. In addition, perception and its incorporation to visualization are derived from psychology-related journals, such as the Journal of Vision, Attention, Perception, and Psychophysics, and Psychonomic. With the survey’s objective and the vast availability of perception-based papers, it was challenging to perform a comprehensive literature search. Our taxonomy discusses the application of perception to visualization. Therefore, I focused on visualization journals and conferences.

I identified papers from major visualization journals and conferences between 1980-2019 (see Table 3.1). I targeted the ACM, IEEE, and EG/CGF libraries to collect the papers using a combination of keywords, including: perception, visualization, evaluation, design, modeling, visual perception, attention, visual task, user study, graphical encoding, and effectiveness. The “others” categories in Table 3.1 are highly cited, pivotal works that were discovered during our search of the primary sources. PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) is an evidence-based minimum set of items for reporting in systematic reviews. PRISMA primarily focuses on the reporting of reviews evaluating the effects of interventions but can also be used as a basis for reporting systematic reviews with objectives other than evaluating interventions. Using the PRISMA framework (see <http://www.prisma-statement.org/>) as a guide, I coded the scope of the survey into categories to screen the literature. I filtered the paper based on the study’s objective and characteristics. Some examples are below.
Table 3.1: The number of surveyed papers by source.

<table>
<thead>
<tr>
<th>Sources</th>
<th>Paper Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE Trans. on Visualization and Computer Graphics (TVCG)</td>
<td>64</td>
</tr>
<tr>
<td>IEEE Information Visualization (InfoVis)</td>
<td></td>
</tr>
<tr>
<td>IEEE Visual Analytics in Science and Technology (VAST)</td>
<td></td>
</tr>
<tr>
<td>IEEE Pacific Visualization Symposium (PacificVis)</td>
<td></td>
</tr>
<tr>
<td>ACM Conf. on Human Factors in Information Systems (CHI)</td>
<td>30</td>
</tr>
<tr>
<td>including Extended Abstracts</td>
<td></td>
</tr>
<tr>
<td>ACM Transaction of Graphics (TOG)</td>
<td></td>
</tr>
<tr>
<td>Computer Graphics Forum (CGF)</td>
<td>17</td>
</tr>
<tr>
<td>Eurographics (EG)</td>
<td></td>
</tr>
<tr>
<td>EG/IEEE VGTC Conference on Visualization (EuroVis)</td>
<td></td>
</tr>
<tr>
<td>including Short Papers</td>
<td></td>
</tr>
<tr>
<td>Others—Beyond Time and Errors on Novel Evaluation Methods for Visualization (BELIV); Journal of Vision; Perception and Psychophysics; Science; Journal of the American Statistical Association; International Conference on Theory and Application of Diagram; Cartographics; Journal of Man-Machine Studies; Behaviour and Information Technology; Others</td>
<td>11</td>
</tr>
</tbody>
</table>

3.1.1.1 Examples for Inclusion

- Experiments focused on graphical perception on visual tasks, e.g., [246], or visual design, e.g., [280].

- Experiments focused on the graphical perception of visualization methods, e.g., [244].

- Studies including discussion and suggestions of design guidelines, e.g., [282].

- Experiments on modeling the visualization to improve inference-making, decision-making, or judgment estimation, based on visual channel and graphical perception, e.g., [120, 330].

3.1.1.2 Examples for Exclusion

- User-study comparing two or more models, e.g., [36].

- Empirical studies to check on quality metrics of a system by user study, e.g., [54].
• Evaluations of the performance of visualization tools or systems, e.g., [321].

• Empirical studies on graphics or user interfaces, not the visualization, e.g., [245].

In addition to the interactive taxonomy at <https://usfdatavisualization.github.io/VisPerceptionSurvey/>., you will find (1) a spreadsheet of surveyed papers, (2) the systematic flow of paper collection and filtering, i.e., PRISMA, (3) a paper categorization template, and (4) a summary table with the count of studies.

3.2 Structure of the Taxonomy

The method of visually encoding data is usually thought to be the main component of visualization. However, the analysis task is equally, if not more, important. Several evaluation studies have suggested visualization effectiveness is task-dependent [244], and a large body of research seeks to determine which data representations are perceptually optimal for specific low-level tasks, e.g., [58, 121, 282, 303]. Seeing that the vast majority of perceptual studies in visualization had a specific low-level task as the main study objective or a low-level task was used in the evaluation, I centered on that as the main category of the taxonomy. Furthermore, most papers consider the low-level tasks in the context of a limited subset of visual encodings and/or visualization types. Therefore, each of the low-level task discussions is further split by types of visual encoding and visualization.

3.2.1 Low-Level Tasks

I considered two existing task taxonomies as a framework for the survey. I first considered Brehmer and Munzner’s taxonomy of abstracted tasks, which is a higher-level taxonomy, i.e.,
perceptual + cognitive [37]. Since the focus was perception, it did not fit well. Despite perception and cognition not being entirely separable, an ideal task framework, would ensure as little cognition as possible occurs within the task. Ultimately, Amar et al.’s low-level task taxonomy [10] fit better, as the tasks they define require less reasoning about the data. From there, I took its ten low-level tasks: retrieve value, filter, compute a derived value, find extremum, sort, determine range, characterize distribution, find anomalies, cluster, and correlate, and I included one derived task, compare, for a total of 11 low-level task groups (see Table 3.2). Each study was placed into one or more of these task groups.

3.2.2 Taxonomy on Visual Encoding

Visual encodings are properties used to encode data in a visualization, including position, length, angle, area, volume, shading, direction, curvature, and color (see Figure 3.3). The terms graphical encoding, visual channel, visual encodings, and visual properties are often interchanged, but generally, they mean the same thing.

![Figure 3.3: Reproduction of visual encodings types in Cleveland and McGill’s work [58].]
Table 3.2: Low-level taxonomy with icon representation and description based on [10].

<table>
<thead>
<tr>
<th>Icon</th>
<th>Low-level Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>📚</td>
<td>Retrieve Value</td>
<td>(section 3.3) – “Given a set of specific cases, find attributes of those cases.” [10]</td>
</tr>
<tr>
<td>🌟</td>
<td>Filter</td>
<td>(section 3.4) – “Given some concrete conditions on attribute values, find data cases satisfying those conditions.” [10]</td>
</tr>
<tr>
<td>🕵️</td>
<td>Compute Derived Value</td>
<td>(section 3.5) – “Given a set of data cases, compute an aggregate numeric representation of those data cases.” [10]</td>
</tr>
<tr>
<td>📈</td>
<td>Find Extremum</td>
<td>(section 3.6) – “Find data cases possessing an extreme value of an attribute over its range within the data set.” [10]</td>
</tr>
<tr>
<td>📜</td>
<td>Sort</td>
<td>(section 3.8) – “Given a set of data cases, rank them according to some ordinal metric.” [10]</td>
</tr>
<tr>
<td>🛠️</td>
<td>Determine Range</td>
<td>(section 3.7) – “Given a set of data cases and an attribute of interest, find the span of values within the set.” [10]</td>
</tr>
<tr>
<td>🌐</td>
<td>Find Anomalies</td>
<td>(section 3.9) – “Identify any anomalies within a given set of data cases concerning a given relationship or expectation, e.g., statistical outliers.” [10]</td>
</tr>
<tr>
<td>🌟</td>
<td>Characterize Distribution</td>
<td>(section 3.10) – “Given a set of data cases and a quantitative attribute of interest, characterize the distribution of that attribute’s values over the set.” [10]</td>
</tr>
<tr>
<td>🌟</td>
<td>Cluster</td>
<td>(section 3.11) – “Given a set of data cases, find clusters of similar attribute values.” [10]</td>
</tr>
<tr>
<td>🧐</td>
<td>Correlate</td>
<td>(section 3.12) – “Given a set of data cases and two attributes, determine useful relationships between the values of those attributes.” [10]</td>
</tr>
<tr>
<td>🎄</td>
<td>Compare</td>
<td>(section 3.13) – “Given a set of data cases, compare any attributes within and between relations of the given set of data cases for a given relationship condition.” [10]</td>
</tr>
</tbody>
</table>

Early works in visual encoding defined numerous individual visual channels, e.g., 10 in [58] and 13 in [177]. Given this taxonomy already defines 11 task categories, enumerating all visual encoding types would have resulted in far too many categories. Instead, I combined visual encodings into roughly two main categories: spatial encodings and color encodings, each with two subcategories (see Table 3.3). Spatial encodings encompass visual encodings having to do with position, size (i.e.,
Table 3.3: Visual encoding taxonomy used in the survey.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Visual Encoding</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Spatial Position and Shape Icon" /></td>
<td>Spatial Position and Shape</td>
<td>Spatial Position and Shape encodings are combined because of their geometric relationship. The category encompasses any encoding concerning the layout, e.g., position, or shape, e.g., direction/angle or curvature.</td>
</tr>
<tr>
<td><img src="image.png" alt="Spatial Size Icon" /></td>
<td>Spatial Size</td>
<td>Spatial Size encodings are those where the size of objects, i.e., length, area, or volume, encode the relevant data in the visualization.</td>
</tr>
<tr>
<td><img src="image.png" alt="Color Hue Icon" /></td>
<td>Color Hue</td>
<td>Color Hue deals primarily with variations in color (in the colloquial sense), usually dealing with categorical colors or colormaps.</td>
</tr>
<tr>
<td><img src="image.png" alt="Color Intensity Icon" /></td>
<td>Color Intensity</td>
<td>Color Intensity encodings capture data by the intensity of the color representation, e.g., in color saturation, luminance, shading, or opacity, which have combined due to their close inter-relatedness, e.g., changes in opacity, luminance, or saturation can all influence the intensity of a color.</td>
</tr>
</tbody>
</table>

length, area, and volume), direction, and shape. Mackinlay, as well as others, have shown that spatial encodings are particularly effective for quantitative data [177]. Color is another important visual encoding that includes properties of color such as hue, saturation, luminance, and opacity. Stimulus-based psychophysics experiments have demonstrated that color can be used to represent ordering [56], category [258], quantity [167], and perceived-difference judgment [280] in visualization.

3.2.3 Visualization

The final category of this taxonomy is visualization type. Most papers studied standard visualization types, quite often with a variation in their design. I extracted all visualization types and combined related types into the following 10 categories and one “other” category represented in Figure 3.4.
3.2.4 Survey

One of the survey’s goals was collecting and framing studies in a way that one could generate effective visual designs considering graphical perception. However, this investigation led me to experiments with limited scope, e.g., focus on a small set of visual encodings, visualization types, limited tasks, or limited data set size and variety. Furthermore, most findings are never replicated or externally validated beyond peer review, with only a few exceptions.

These limitations have two effects. First, when the scope of work is limited, it puts transferability of findings on unsteady ground. For example, if the real-world conditions associated with a data visualization do not match a contrived scenario, often necessary for studying graphical perception, will the findings be relevant? The second limitation is that finding common links between various studies is quite often difficult. For example, if two studies using different sized data and slightly different tasks have contradictory results, relating the findings in an actionable way is difficult at best. As such, summary of works in graphical perception tries to highlight these differences while leaving room for the reader to interpret real-world consequences for their particular situation.
In the coming sections, the taxonomy of each task consists of the task description, findings per visual encoding, and visualization concluded with a summary.

3.2.5 Discussion

One of this survey’s goals is collecting and organizing findings so that the reader can make a judgment of the applicability of results to their context. Based on the systematic review of research in the perception of visualizations, their findings, and taxonomy, I demonstrated the potential area of study. I concluded our report (section 3.14) with a summary of the weaknesses and open research questions in the area.

3.3 Retrieve Value

The retrieve value task requires identifying the data or attributes that satisfy a given set of specific criteria. In the literature, retrieve value tasks are often combined with other tasks, e.g., computing a derived value, comparison, sort, etc. I did identify several works that studied the retrieve value task individually with various visualization methods.

3.3.1 Visual Encoding

3.3.1.1 Spatial Position, Shape, and Size

Visual factors, such as position along common and unaligned scales, have been observed to produce more accurate judgments than length, direction, and angle on simple tasks [58, 177]. However, a recent study demonstrated in position-based visualizations, e.g., scatterplots, encoding additional attributes with size required more time for retrieving values than alternatively encoding with color [147]. On the other hand, subjects’ accuracy improved with size. The findings further
explored the symmetry between the size and color as the visual encoding for quantitative information, and they suggested that marks with varied sizes might interfere with decoding the quantity value in position channels. The results further indicated that chart orientation influenced performance by comparing x- and y-faceted charts, with y-faceted charts performing 0.9 times faster. Nevertheless, our understanding of how variations in the visualization design potentially influence user performance on this task is evolving.

3.3.1.2 Color Hue

Several studies have evaluated people’s ability to identify values using color-coded visualizations. In one instance, the performance in identifying or determining value in visualization was shown to be affected by color and other visual features, such as motion or layout [118]. I have identified several studies focused on how color can be applied such that finding targets or reading a value becomes less strenuous. For example, the issue of color vision deficiency, i.e., colorblindness, was studied in automated colormap design [289]. In the study, colormaps were created starting with a single seed color that was used to generate color ramps that mimic designer practices. Their experiments measured the viewer’s potential to accurately identify or read a value in the given color-coded scatterplot, heatmap, or choropleth maps. Ultimately, the performance of their automated system was as good as human-designer-specified color ramps [268]. Reasonable precaution is necessary to improve accessibility, i.e., for colorblind, low vision, or other vision impairments, to make the task less challenging and provide an unbiased user response.
### Figure 3.5: A study evaluated a pairwise relation between visualization types across tasks. Arrows show that the source is significantly better than the target in terms of performance metrics. Image reproduced with permission [244].

#### 3.3.1.3 Color Intensity

One study showed that lightness in the data point symbol could be an impacting factor for visual tasks, such as locating and identifying values in sparse scatterplots on a white background [164]. This experiment identified how lightness could be modeled as a combination of two opposite power functions and used to determine discriminability. Related to lightness, opacity has been used as a constructive factor in a scatterplot design to mitigate overdraw in tasks, such as to read and identify values for higher-level tasks [172].
3.3.2 Visualizations

A recent study by Saket et al. [246] comparing five visualization types—tabular visualization, scatterplots, bar charts, line charts, and pie chart—evaluated the efficacy of these visualizations with small data for all ten of Amar et al.’s tasks [10], in terms of accuracy, time, and user preference (see Figure 3.5). For the retrieve value task, they found tabular visualization outperformed all others for accuracy, time, and preference, bar and pie charts performed well in terms of accuracy and time, and scatterplots performed well only in terms of accuracy. A similar comparative study of three multidimensional visualizations—parallel coordinates, scatterplot matrix, and tabular visualization—studied their user performance on analytical and decision-making tasks. For the retrieve value task, parallel coordinates had the highest accuracy. The evaluation also demonstrated that tabular visualization was familiar, accurate, and time-efficient for the retrieve value task [75]. In both studies, tabular visualizations allowed subjects to reach decisions faster with better accuracy levels than the other visualizations because of their familiarity with tables.

3.3.2.1 Scatterplot

A scatterplot is one of the most effective forms of visualization, which allows users to identify, read, or retrieve a value based on different visual encodings. For example, scatterplots colored with additional categorical data were found to be highly effective for comparing individual data and were preferred over dot plots [147].

3.3.2.2 Bar Chart

While bar charts represent information in a way that helps the user identify and read quantitative data easily, the design decisions used play an essential role in user performance. In one
study, pictorial bar charts were found to reduce the user performance on retrieving a value task but not beyond the effect already observed based upon their shape [266]. A recent study focusing on a “reading a value” task concluded that there is a need for direct encoding of absolute values and their relationship in bar charts or dot plots [197]. Ultimately, while embellishments are used to improve aesthetics and memorability, they can affect user performance on the task.

3.3.2.3 Parallel Coordinates

Parallel coordinates are often disparaged as being challenging to use and understand for first-time users. However, one study used the retrieving a value task to show otherwise—the experiment assessed the task using eye-tracking, and it indicated that first-time users quickly learned to use them [263]. The study plotted eight vehicle attributes as axes for 406 cars and used eye-tracking to measure the user’s performance on identifying the values in the data.

3.3.2.4 Text

In visualizations where text is involved, visual complexity affects a user’s ability to grasp aspects of the overall structure of the visual display. Optimizing the use of typography in visualizations, e.g., the location of the labeling and parameters including typeface, font size, font weight, color, orientation, intensity (boldness), spacing, case, border, background, underline, and shadow, determines the legibility of text and, thus, influences the understandability of the visualization [276]. One experiment indicated the biases caused by word length, height, and width could impact user accuracy, and the findings provide practical guidelines for improving the user experience [6]. Therefore, the design of text in a visualization should consider the word typography for any potential biases.
3.3.2.5 Map

We use larger and higher resolution displays to increase the scalability of visualizations, particularly when data is small, such as in maps. By sheer size, such displays could make it more difficult to attend to the visualization. A perceptual scalability study on maps investigated user performance focusing on retrieving or reading a value task in a larger display [333]. They found that the larger display did *not* result in a time increase or accuracy decrease.

In summary, I identified several studies using the retrieve value task scattered over various visualization types. As one would expect, user performance varies with the choice of visualization and the visualization design. Prior work indicates that bar charts should be the first choice when accuracy is important. Thoughtful use of embellishments can improve aesthetics and memorability, but they can also affect user performance on the task.

3.4 Filter

In our search of the literature, filtering essentially came down to two types of tasks. The first, what we would typically call filtering, encompassed eliminating subsets of data used in the visualization. The second was a search task, which focused on finding a target in the visualization.

3.4.1 Visual Encoding

3.4.1.1 Spatial Position, Shape, and Size

Popout, a pre-attentive effect caused by variations in certain encodings, makes it easy to draw the user’s attention to the critical elements of the visualization. A study examined and demonstrated the efficacy of popout in target identification in a group of symbols in a scatterplot using different
visual channels, including color, shape, luminance, flashing, motion, and size [115]. On three metrics of performance—perceived success, visibility, and accuracy—shape required high effort with a low performance, whereas motion showed little effort and a high level of performance. The remaining four channels showed a definite increase in perceived visibility and accuracy across intensity.

A similar experiment investigated searching for a target in a grid matrix using three visual channels—mark size, set size, and color—suggested design guidelines based on grouping, quantity set, and size of visual symbols [111]. While searching for an item was faster when colors were spatially grouped in the grid, the number of symbols had little effect on search time. At the same time, if the number of symbols increased, the performance slowed for random data displays.

3.4.1.2 Color Hue

Color encoding optimization can increase the effectiveness of categorical data visualization. While categorical colors are easily distinguishable in large color bars or individual plots, when they are inserted into maps or other plots with varying sizes, the variation reduces the visibility of categorical differences. Class visibility is an important measure of how color and spatial distribution of each class affect its perceptual intensity to the human visual system [160].

3.4.2 Visualizations

Filtering task performance highlights the need to explicitly and directly encode numeric differences between data values [197]. In an experiment with image visualization, finding an image within a set of images showed that both latency and task complexity play a significant role in search behavior [18]. In Saket et al.’s study on small data [244], they found tabular visualizations and bar charts performed well in terms of accuracy, time, and preference on filtering tasks (see Figure 3.5).
3.4.2.1 Parallel Coordinates

In a decade-old experiment, a filtering task was used to assess the usability of parallel coordinates. The performance of the filtering task using eye-tracking showed that users had high fixation time in order to confirm their interpretation of the results [263]. The study showed that the effect of latency in an interactive visual system is more gradual than binary.

3.4.2.2 Map

Maps are used to represent structures, patterns, and relations in spatial data. Researchers have worked on a variety of map types, e.g., choropleth [21], cartogram [199], geographical map [160], typographic maps [3], etc., to evaluate the human perception in identifying the pattern and structures, and optimize the design process based on the encoding method. As pointed out earlier, class visibility has been studied on maps and shown how color and the spatial distribution can influence perceptual intensity [160]. Filter and search tasks on maps are affected by visual encoding choices, as well as the physical construction of maps. Cartograms were evaluated on the four classes—contiguous, non-contiguous, rectangular, and Dorling—using a qualitative performance analysis [199]. Cartograms that preserve the relative position of the regions in the geographical maps, facilitating faster search, while Dorling and rectangular maps had better accuracy.

3.4.2.3 Text

The optimal setting of typography parameters determines the legibility of text, thus, influences the understandability of visualizations. In a word cloud study focused on different layout and encodings effects, font size and color performed well compared to encodings using additional marks, such as bars and circles on keywords, in a task of searching for a keyword [100] (see Figure 3.6).
Figure 3.6: Examples of text visualizations generated by visual encodings and layouts[100].

Different representations affect people’s performance in extracting information from text visualizations, such as word clouds. This design space summary shows examples of visualizations generated by different visual encodings and layouts. Image reproduced with permission [100].

Furthermore, spatial and column layout of the keywords similarly outperformed the row version of keyword arrangement on keyword searches. A similar study on text confirmed the prior results by showing how searching for target words could be influenced by font sizes [6].

Another study evaluated the effects of four types of layouts, and the results showed that perhaps unsurprisingly, alphabetic ordering was faster for searching for a keyword, order and font size were found to have no impact on searching for a tag belonging to an assigned topic, and words with bigger fonts were more likely to be recalled [251]. This confirmed a prior study that investigated word recall based on classic word clouds with unordered layouts and font size encoding frequency. In a study, participants recalled the words with larger font sizes more often. Their guidelines considered optimizations based on typography (i.e., font weight, size, and color) and word placement (i.e., sorting, clustering, and spatial layout) [233]. In addition to layout and font size, good highlighting mechanisms facilitate search and comparison in textual views or labels. Another study empirically investigated the effective use of highlighting techniques for visualization applications for
text data and suggested design guidelines for the effectiveness of nine web-friendly text highlighting techniques [276]. Search tasks on the text and word clouds benefit from the thoughtful selection of size, layout, ordering, and highlight.

In summary, the filter task is one in which user performance varies based on the design of visualization, choice of visualization, and related set of tasks being performed. Tabular visualizations and bar charts perform well in terms of accuracy, time, and preference. Searching for a keyword in text or a value in a scatterplot can benefit from utilizing encodings font/symbol size and color, as well as alphabetic ordering for text.

3.5 Compute Derived Value

Given a set of data, computing derived values is similar to calculating an aggregate of that data. Many visualization tasks require users to create an aggregated abstraction or statistical summary, which are often referred to as visual aggregations.

3.5.1 Visual Encoding

3.5.1.1 Spatial Position, Shape, and Size

Cleveland and McGill’s seminal work on visual perception utilized the task of computing derived values from the visual representation [58]. In their results, position and length were among the visual encodings with the highest accuracy in quantitative judgment. That study was replicated on a crowdsourced platform using proportional judgment tasks to confirm the effectiveness rankings of visual encoding [123]. A recent study, concentrating on various channels in scatterplots, suggested point size encoding performs well for summary tasks on quantitative encoding [147].
Another experimental study investigated the estimation of average position in line and bar charts [327]. Even though positions are considered a precise form of data encoding, reports of average positions were biased because of underestimation or overestimation of bar positions due to the introduction of a bias called *perceptual pull*—position estimates for a target data were “pulled” toward the irrelevant data in the series.

### 3.5.1.2 Color Hue and Intensity

Quite often, colormaps are applied to data, and aggregation tasks are performed. Studies have demonstrated the perception of continuous colormaps is affected by colormap characteristics and spatial frequency. Estimating the values based on colormaps in a continuous quantitative colormap showed no relation between colormaps and spatial frequency, but with increased spatial complexity, estimation error increases [226]. The important point is that spatial frequency impacts the effectiveness of color encoding, but the true impact is task-dependent.

Similarly, another study assessed the efficacy of colormaps for encoding scalar information using binary-choice experiments [169]. Relative distance judgments investigated accessing color similarity between the source and target using reference color. A combination of perceptually uniform color space and color naming more accurately predicted user performance than either alone, but the accuracy was low in both cases.

A study, which used color as an encoding method on time-series data was performed to identify the average in data, and it demonstrated that color-coding techniques showed better average judgments [65]. Symbols are used in a scatterplot to encode values, and lightness is one of the dominant factors of color encoding. A model-based study on computing derived values compared two
different types of scatterplots—with a circle as symbols and with spots as a symbol—with varying levels of lightness [164]. The scatterplot with a circle as a symbol performed better in terms of error rate. Gleicher et al.’s large-scale crowdsourced study on identifying aggregate relative judgment to read and the average value in the multi-class scatterplot was another form of computing derived values on various groups of objects [109].

3.5.1.3 Other

Factors, such as affective priming, social information, or individual user characteristics, directly impact the task performance. Accurate visual judgment is essential to performing the summarization, estimation, or other related computations of a derived value. Harrison et al. performed a series of experiments that showed users articles from the New York Times, followed by a judgment task [120]. The evidence from the experiment showed that affective priming influences the user’s visual judgment accuracy. In similar experiments, social information was shown to influence the users on summarization tasks, such as proportion judgment and linear estimation [129]. Other studies have evaluated how individual user characteristics, in particular, working memory (WM) (low, average, or medium), affected the user’s performance on visual tasks, such as counting values [61]. The experiment evaluated the task by comparing charts with vertical and horizontal layouts. Users with low visual WM performed faster with a horizontal layout.

3.5.2 Visualization

When comparing the overall efficacy of different visualization methods for computing derived values, Saket et al. showed that for small data sizes: in terms of accuracy table, bar chart, pie chart,
and scatterplots were best, in that order; in terms of completion time table and pie chart were best; and finally in terms of user preference, table, pie chart, and bar chart were best [244] (see Figure 3.5).

3.5.2.1 Bar Chart

People prefer linear bar charts because of their familiarity. A comparison-based study between using radial and linear bar charts to visualize daily patterns suggested that 24-hour linear bar charts are more accurate and efficient for summarization tasks [298]. Along with the layout of bars in a bar chart, the design of the bars themselves also affects user performance. In a recent study on comparison tasks on various types of bar charts, two types of compute a derived value task were studied to identify the effects of bar variants on tasks [272]. They found chart design affects the task completion time, and data conditions also influenced the completion time and accuracy.

3D bar graphs are considered to be difficult to understand and generally bad for performing visual analysis tasks. In an experiment where the performance of relative magnitude estimation on pie charts and bar charts with and without 3D was evaluated, the task performance was shown to be better in 2D than 3D [262].

As better design improves performance, visual embellishments are often used to improve aesthetics. Embellishments were evaluated in bar charts on relative magnitude estimations, which confirmed that common embellishments significantly impact the task [264]. In another follow-up study, pictorial charts reduced the user’s performance on relative judgments or computing a retrieved value task, but not beyond the effect already observed for their shape [266].
In an evaluation of pie and donut charts on relative magnitude estimation performance, the baseline donut chart was observed to be as good as the baseline pie chart. Furthermore, user performance on the arc length chart was similar to an area chart, angle pie chart, and angle donut chart [265]. In a similar study, where judgment error in pie chart variations was evaluated, variants of the pie charts, such as a chart with a larger slice, exploded pie, elliptical pie, and square pie caused more significant judgment errors than regular pie charts [26].

A study compared different types of cartograms on tasks, such as detecting change and summarizing, found that contiguous cartograms performed better at detecting change, with the lowest error rate and completion time. For the summarize task, contiguous and Dorling cartograms had lower completion times, while the error rate in rectangular cartograms was highest among all four types of map visualization [199].

In summary, computing derived values is one of the more basic low-level tasks, generally studied as a standalone task. User performance for this task varies greatly based upon visualization and encoding type. Though, the majority of the studies considered only bar charts and line charts. Overall, prior studies have indicated that bar charts, line charts, pie charts, and scatterplots can all be effectively used for summary-based task visualizations.
3.6 Find Extremum

This task focuses on identifying data cases that possess an extreme value of an attribute over its range within the dataset. In this context, the find extremum values task can encompass finding both global and local maxima or minima in data.

3.6.1 Visual Encoding

Position encoded plots, e.g., scatterplots and dot plots, are often used to identify quantitative values [58]. Further, in a study of the effectiveness of visual encodings, the size of marks was more effective than color for quantitative values involving finding or identifying the extremum values [147].

3.6.2 Visualization

For the find extremum task, Saket et al. found all five methods—tabular visualization, scatterplots, bar charts, pie charts, and line charts—performed well in terms of accuracy on small data. Bar charts, line charts, and scatterplots additionally performed well in terms of time. Finally, bar charts performed best in terms of user preference [244] (see Figure 3.5). Another recent work compared the performance of design variation in three visualizations—bar charts, line charts, and pie charts—on a task of finding the maximum [201]. The evaluation confirmed the prior hypothesis of better performance in overlaid design versus small multiples and suggested additional new performance improvements.

3.6.2.1 Bar Chart

Bar charts are often used to read the minimum/maximum quantitative values in data, and recent studies have suggested have indicated that bar design in bar charts influences the user’s
performance on identifying those values [272]. In another example, a comparison between radial charts and linear bar charts investigated the visualization of 24-hours time. Users had higher accuracy for identifying the maximum value in linear bar charts compared to radial charts [298]. A series of studies were performed for determining the maximum value of delta and mean in bar charts, slope, and donut chart for three variations of their design—small multiples, overlaid, and mirrored [132, 201]. Animation in the design made identifying maximum delta value particularly salient, but the effect did not carry over when the change was large. The identification of maximum mean value held high accuracy for the bar charts with mirrored and stacked bar arrangements.

3.6.2.2 Line Chart

Line charts are used to visualize time-series data because of the inherent potential of showing the trend. Evaluation of variation in line graph design demonstrates user performance of the finding minimum or maximum values also depends on the design of the visual representation [5]. A modified form of line charts using position encoding, called a stock chart, performed significantly better on finding a minimum, whereas the composite line chart performed well on finding a maximum. However, line charts using a colorfield had the highest accuracy on this task.

3.6.2.3 Network and Map

A comparison study between node-link graphs and matrix-based representation reported that the task of “finding extremum,” in the form of finding the most connected node, had the highest

---

2 The modified stock chart is a line chart with a layering of moving average over the original series to supplement summary judgments [5].

3 A composite line chart is a line chart layered over a bar chart representing averages of discrete subregions [5].
accuracy and lower completion time for a matrix representation [107]. The results were particularly strong as the size of the graph became large, or the link density increased.

The task of identifying the highest value of an attribute in maps, in the form of top-k identification, a standard cartogram has the highest accuracy when compared to other types of cartograms, i.e., rectangular, non-continuous, or Dorling cartograms [199].

In summary, finding extremum values is a low-level task mostly studied on bar charts and line charts. Generally speaking, line charts can be used to represent the time-series data when finding an extremum value, while scatterplots and bar charts can be used for quantitative values encoded by position or length.

3.7 Determine Range

A determine range analysis task has users finding the span of values in a given data for an attribute of interest. The determine range task is another task that has received limited attention.

3.7.1 Visualization

Saket et al.’s evaluation on small data found that for the task of determining a range of values, bar charts had high accuracy, whereas scatterplot performed better on completion time and user preference [244] (see Figure 3.5). In an evaluation of multidimensional visualizations, where the task of finding range is evaluated on three user performance metrics of accuracy, completion time, and user satisfaction, parallel coordinates and tabular visualization were found to be effective in terms of accuracy and completion time, but parallel coordinates were least preferred by users [75]. However, this result is only applicable to this limited context.
3.7.1.1 Bar Chart

A variety of bar charts arrangements designs were investigated on a maximum range task [132]. A stacked arrangement of bar charts gave the highest visual comparison accuracy, while superimposed charts gave the lowest.

3.7.1.2 Line Chart

For identifying the range of values on time-series data visualizations, line charts based on position encoding had higher accuracy over a modified stock chart, box plot, and a composite line graph, respectively [5]. Though these are variations of the same visualization type, i.e., a line chart, their design variations significantly impact their efficacy.

In summary, determining a range of values has many unstudied aspects. For example, I did not find any studies which directly investigate the performance on visual encodings. That said, the limited depth and breadth still indicate that scatterplots are preferred when requiring faster performance, bar charts should be chosen when accuracy is needed, and parallel coordinates work well for multidimensional data.

3.8 Sort

Sorting generally implies ranking the given set of data according to some ordinal attribute. Sorting, as a low-level task, has not received much attention.
3.8.1 Visualization

Saket et al. also studied ordering tasks, which are synonymous with sorting [244] (see Figure 3.5). The bar chart stood out in terms of accuracy, timely completion, and user preference. Pie chart, scatterplot, and tabular visualization were next best in terms of accuracy, whereas the line chart and scatterplot were next best for the time completion metric.

3.8.1.1 Bar Chart

Different types of ranked-list visualizations, such as scrolled bar charts, wrapped bars, piled bars, packed bars, treemaps, and Zvinca plots, were evaluated in a study that suggested wrapped bars are best for visualizing ranked lists as they provide a simple, compact, and interaction friendly visualization, while treemaps reported the highest accuracy despite the use of area when length is generally a preferable encoding [189].

3.8.1.2 Parallel Coordinates

An improved form of parallel coordinates, called the progressive parallel coordinates, was studied to understand how data order can mitigate scalability concerns [237]. The application of level-of-detail and randomly accessing individual values were based on an ordering activity.

3.8.1.3 Text

Sorting a keyword summary alphabetically reduces the time to find a word, and as such, it makes ordered layouts (e.g., column layouts) more effective than unordered layouts (e.g., spatial arrangements) [116]. It was shown that a simple ordering of text data could be much more effective

---

4 A Zvinca plot is a layered plot where points replace bars, see [102].
than font size encodings, most notably in searching and retention. In other cases, words can be sorted alphabetically, by frequency, or by a predetermined algorithm. In an experiment on word cloud effectiveness suggesting guidelines for construction, the authors evaluated impression formation and memory by varying font size and layout (e.g., alphabetical sorting, frequency sorting) of words [233]. One important finding was that a list ordered by frequency might provide a more accurate impression as compared to other layouts.

3.8.1.4 Other

A study focused on user characteristics showed that user performance on sorting values depended upon participants’ working memory (WM) (low, average, and medium) [61]. Due to individual differences in perceptual judgments, users with lower verbal WM were slower than others for the sorting task, and users with low visual WM required more time for the task than users with average visual WM. Users with higher cognitive processing speed were also more effective at deriving facts and insights from a visualization than others.

In another study on perceptual kernels—a matrix of aggregated pairwise subjective measures of judged similarity—participants were asked to rank the data categories from least to most similar to a target class [71]. This experiment estimated perceptual kernels for visual encoding variables of shape, size, color, and combinations. Based on the judged similarities using Likert ratings among visual variables, findings can be applied to improve visualization design through automatic palette optimization.
In summary, the sorting, which has many similarities to finding extremum and determining range, has not been studied extensively. Nevertheless, studies indicate that bar charts and text-based visualizations are the preferred techniques for quantitative and textual data, respectively.

3.9 Find Anomalies

The find anomalies task is a form of visual aggregation task, generally involving identifying any outliers or unexpected cases within a given set of data. Finding anomalies is also, which is defined by the need to identify targets that are different from others in the given set.

3.9.1 Visual Encoding

3.9.1.1 Spatial Size

With the focus on investigating symbol size discrimination in scatterplots, one study developed a method to pick sizes that are effective for counting outliers and making the judgment as easy as possible [163]. The results show that size perception can be described by the Power law transformation to yield an optimal scale for symbol size discrimination.

3.9.1.2 Color Hue

Color, motion, and layout (random or grouped) affect the users’ attention capacity, and the same has been evaluated on identifying outliers [118]. Search time for an outlier in a matrix form of symbols was shown to vary with color, motion, and layout. Further, color grouped and motion grouped had the best response time, whereas motion random cases were worst in response time. The color group also had the highest identification accuracy.
3.9.1.3 Color Intensity

Another discriminability study aimed to select luminance levels such that analytical tasks, including counting outliers, were as easy as possible [164]. The performance limit for the task was approached where high-perceived contrast was observed for the data symbols. Additional findings stated that with clustering/groupings of the same data symbols, the task might become easy for low contrast sets.

3.9.2 Visualization

Saket et al. found that for finding anomalies, scatterplots and bar charts performed best in terms of accuracy, time, and user preference [244] (see Figure 3.5).

3.9.2.1 Line Chart

Identification of outliers in time-series data using line charts and their variants is complex and reported low user response accuracy [5]. Basic line chart, modified stock chart, composite line chart, and color-field have the same level of accuracy, whereas event stripping outperformed all of these visualizations. The accuracy was low in this visualization for outlier tasks because data spread was confounding as an outlier and vice-versa. Essentially, when data values that would increase or decrease in the opposite direction were focused on for outlier behavior [331].

3.9.2.2 Other

A recent study on three types of visualization—density plot, histogram, and dot plot—pointed out the outlier detection task as a part of data quality issues [67]. Participants were better able to
identify outliers as the flaws in data, but there was no single visualization suggested, which was significantly better among the three.

In an algorithm-based outlier detection, the authors studied outliers and anomalies on box plots and letter-value-box plots [316]. These types of approaches can be paired with visualizations to help analysts explore anomalous data features, especially multidimensional outliers.

In summary, finding anomalies is defined by the need to identify targets that are different from others in the given set, which varies based upon the visual features, e.g., position, size, orientation, color, and luminance [281]. Scatterplots have been most heavily studied for this task. They are efficient in identifying outliers and detect anomalies in the data. Additionally, line charts easily represent any abnormal or outliers behavior in time-series data.

3.10 Characterize Distribution

This task requires that for a given set of data and a quantitative attribute of interest, the distribution of that data should be characterized over that attribute’s value.

3.10.1 Visual Encoding

Characterizing distribution is another visual abstraction task, with a specific focus on pattern or trend recognition. The data can be encoded using several visual features, e.g., position, size, orientation, color, and luminance, which can be crucial in identifying the distribution [281]. For example, Gapminder uses size, position, and color to reveal patterns in global demographics [239], and weather maps use color and orientation to visualize information about wind speed, temperature, and other meteorological data [304].
3.10.1.1 Color Hue

Colormap design in continuous quantitative maps was used to evaluate the perception of patterns [226]. There was a negative main effect of spatial frequency on pattern perception. Further, the use of low color at low spatial frequency was unlikely to improve pattern perception as compared to a plain grayscale ramp.

3.10.2 Visualization

In Saket et al.’s study, the bar chart was shown to have high accuracy and to be the user preferred method for characterizing distribution tasks, but the scatterplot had better completion time [244] (see Figure 3.5). The line chart was next most accurate after the bar chart and scatterplot.

3.10.2.1 Scatterplot

The scatterplot uses position encoding for the data, and users can identify or detect a data pattern or distribution easily. Varying densities and gaps between data points were shown to influence the user’s perception of the distribution or pattern of data points [200]. Further, multiple studies demonstrated the task of characterizing a distribution is influenced by encodings, specifically symbol size and lightness, in a scatterplot [163, 164].

With scagnostics, density is assumed to be a property that shows the concentration of points, which is directly influenced by the distribution of points [317]. This observation led the way to investigating how people interpret trends in a scatterplot and was studied using a sensitivity model. Visual augmentation of scatterplots introduces sensitivity information, which was used to study how people interpret trends in scatterplots [45]. Orientation cues provided by the flow lines give an idea
of the local data. Sensitivity or local trends helped in identifying the type of relationship between two variables.

3.10.2.2 Line Chart

Variants of the line chart were evaluated on characterizing the distribution of data, also called the spread [5]. When encodings are position-based, the box plot performed with the highest accuracy, but in the case of color encoding, event stripping performed better. Conventional line charts stood third in the ordering after bar charts and scatterplots for pattern identification. While line charts are usually considered to be the best choice for time series visualization, scatterplots are more effective for showing trends. A study was conducted to merge the two, automatically selecting the right representation for trend exploration in time series data [301]. The choice was affected by the amount of noise, outliers in the data, and aspect ratio. When the noise was small, a line chart is preferred, whereas a scatterplot was preferred with the larger noise.

3.10.2.3 Parallel Coordinates

A new technique was combined with parallel coordinate, called orientation-enhanced parallel coordinates, to overcome the clutter due to overplotting for characterizing the underlying data distribution [223]. This method improves outlier discernibility by visually enhancing parts of each parallel coordinates polyline through its slope. Interactive evaluation verified the feature of the discernibility of information in complex data.
3.10.2.4 Maps

Cartograms are popular for geo-referenced data visualizations used to illustrated patterns and trends in the map. Major types of cartograms (e.g., contiguous, non-contiguous, rectangular, Dorling, etc.) were evaluated for comparing trends and analyzed on quantitative performance analysis in terms of completion time and error and subjective preferences [199]. Dorling cartograms performed best on the task involving comparing trends, whereas rectangular performed worst.

3.10.2.5 Other

Viewers accurately estimate trends in many standard visualizations of bi-variate data. However, visual features, e.g., bias within a bar of visualization, and data features, e.g., outliers, can result in visual estimates that systematically diverge from standard least-squares regression models [66]. Designers should be aware of the distinction between regression by eye and explicit statistical information.

Scatterplots and line charts have received the majority of attention for the task of characterizing the data distribution. Scatterplots support faster and easier identification of distributions and patterns in data, followed by line charts. A line chart should still be used when the patterns or trends are from time-varying data. Pattern perception on colormaps or chloropleth maps should be sure not to use low color combined with lower spatial frequency. Otherwise, performance will suffer.

3.11 Cluster

Clustering tasks are focused on identifying a similar attribute in a given set of data. A design factor survey study on information visualization defines clustering as a high-level data
characterization—“the ability to identify groups of similar items” [248]. Clustering and segmentation of data points in a given dataset reveal characteristics of data and allow visualization designers and practitioners to explore more about the data [247].

3.11.1 Visual Encoding

3.11.1.1 Spatial Position, Shape, and Size

Symbol or mark size is an influencing visual encoding that affects the density and concentration of point clustering. Additionally, symbol size has a direct influence on identifying the clusters in data, and studies have demonstrated that their discriminability is task-dependent. Li et al. demonstrated the effect of mark size and lightness perception on viewers’ ability in multi-class scatterplots for clustering-based tasks [163, 164]. Separability between the symbols or groups of symbols is an important factor in identifying clusters based on the encoding marks. For example, mark shape significantly affects the perception of both size and color, and separability among the three encodings function asymmetrically [267].

Since concentrations of density influence cluster perception, as the size of data points increases, so does the concentration and density. Sadahiro developed a mathematical model to represent cluster perception in point distributions based on proximity, concentration, and density change, and he suggested perception was significantly influenced by the concentration and density change [242]. Varying densities and gaps between groups of points influence the pattern of data points, potentially forming clusters of points [200]. A study focusing on the perceptual optimization of scatterplot design studied standard design parameters, including mark size, opacity, and aspect ratio, and it demonstrated that effective choices of the variables enhanced class separation [186].
3.11.1.2 Color Hue

Color is another important visual encoding of a scatterplot that influences the visual task of segmentation or clustering and grouping, but how we measure the color difference perception varies inversely with mark diameter [280] (see Figure 3.7). Furthermore, the shape of the symbol significantly affects how well we perceive the color difference [267]. Hence, an optimized choice of colors aids users in efficiently understanding separability in multi-class scatterplots. They used a method of color assignment to design scatterplots that optimized class separability perception taking into account density-related factors, such as spatial relationship, density, degree of overlaps between points and cluster, and background color [300], which could not be achieved by the default colormapping.

3.11.1.3 Color Intensity

Reducing mark opacity can alleviate overplotting to aid various visual analytics tasks, e.g., cluster perception or identification [180, 220, 247] while preserving the spatial information. Different opacity levels aid in enhancing class separation—while low opacity benefits density estimation for extensive data, it also makes locating outliers more difficult [186]. Of course, these solutions have their limits, e.g., when opacity is below available precision, or points are their smallest possible size [103].

3.11.1.4 Bias

Priming and anchoring effect distorts the user’s decision-making process. Deciding the separability of the two clusters depends not only on how far they are apart but also on previously seen stimuli. Valdez et al. elaborated on how repeated exposure to a visualization impacts our
interpretation [294]. While the authors presented the biases caused by these effects, at the same time, the results came with the caveat that judgments were not normally distributed, which further indicating an overestimation of the effect size. The effects can be accidentally caused by chromostereopsis [8] and should be studied on monochrome colors.

3.11.2 Visualization

Saket et al.’s study found that, with small data, bar charts and pie charts outperformed tables, scatterplots, and line charts in clustering tasks [244] (see Figure 3.5). The performance in cluster perception in pie charts can be traced back to its effectiveness in facilitating proportional judgments through a part-to-whole relationship [87, 271].

3.11.2.1 Scatterplot

Scatterplots are widely used to visualize data to reveal patterns of data characteristics such as class segmentation or clusters. Numerous studies on visual cluster separation have been conducted to identify the effects of various design factors and visual encodings.

A qualitative evaluation of cluster separation measures study suggested taxonomy of four factors—scale, point distance, shape, and position—that influence separation perception [256]. With the focus class separation, Sedlmair and Aupetit’s extended their prior work and evaluated 15 state-of-the-art class separation measures in a study aimed at mitigating human judgment impacts. They rely on human ground truth as input to a machine learning framework that was used for evaluating the quality of dimension reduction [254]. A further continuation of the work included even more measures for improved matching to human perception [15]. The ScatterNet method captured perceptual similarities between scatterplots by applying a deep learning model that was designed
Figure 3.7: Perceived color difference varies inversely with size, and elongated marks [280]. Colors on longer marks were also more discernible than shorter bars of equal thickness. The perceptible color differences for lines vary inversely with thickness. Finally, perceptible color differences for points vary inversely with point diameter. Image reproduced with permission [280].

to emulate human clustering decisions [176]. The scagnostics technique focused on identifying the patterns in scatterplots, including clusters [69, 181]. However, a recent study showed that scagnostics do not reliably reproduce human judgments [206]. The commonality in all of these studies is that they are algorithmically oriented, and most of their evaluations did not consider visual channels.

Sadahiro presented a mathematical model that suggested perception is significantly influenced by the concentration and density change [242]. ClustMe used visual quality measures to model human judgments to rank scatterplots [1]. ClustMe performed well in reproducing rational decisions for cluster patterns. Quadri and Rosen built and tested a topology-based model of human perception of scatterplots that considered data distribution, number of data points, size of data points, and opacity of data points in cluster perception [220]. Their model demonstrated the strong relationship between all of these factors and the perception of cluster separation.

3.11.2.2 Parallel Coordinates

Parallel coordinates are useful for a variety of tasks, including clustering in real-world applications [195]. One study that measured the participants’ learning outcomes used clustering
as a primary task [155]. The results showed a more engaging experience for interactive parallel coordinates than static, and the users did not find it challenging to learn the data item mapping to the parallel axes.

Orientation-enhanced parallel coordinates were developed especially for large datasets to improve the display of emphasizing the underlying data structure, such as allowing the discernibility of clusters [223]. The evaluation demonstrated that orientation made it easier to identify the data clusters between the two data dimensions and allowed multiple small clusters between the first two dimensions to be visually enhanced. Ultimately, there remains some question as to whether scatterplots or parallel coordinates are better for identifying clusters, as one study found that parallel coordinates better showed the actual shape of clusters [126].

3.11.2.3 Networks

Graph layout algorithms optimize visual characteristics of visual encodings to create intuitive visualizations. The layout of graphs was investigated in an evaluation study where participants produce their graph from a graph shown earlier to depict clusters [295]. Users reach confidence and higher overall task accuracy in visualization during the interaction when rapidly adjusting the visual encodings.

3.11.2.4 Text

A study to investigate word recall memory, based on classic word clouds layout and font size, stated that word placement (i.e., sorting, clustering, spatial arrangement) is an important consideration as it affects the users' recall potential [233].
As scatterplots demonstrate the clustering of bivariate data effectively, most studies I identified used scatterplots. In the case of multivariate data, parallel coordinates and scatterplot matrices can be used to visualize data.

3.12 Correlate

Correlation, in general, is a relationship between values of two or more attributes in a dataset. Formally, correlation is a statistical measure of the linear relationship between two quantitative variables, represented by a correlation coefficient. A positive correlation indicates the extent to which those variables increase in parallel, and a negative correlation indicates the extent to which one variable increases as the other decreases.

3.12.1 Visual Encoding

Evaluating the correlation perception helps identify people’s abilities to perceive and judge differences in visualizations. Numerous studies have focused on identifying linear correlation in a data distribution with visual encodings, including the slope of the points, marker size, opacity, and color [57, 121, 162, 186, 230]. Rensink and Baldridge showed that perception of correlation in scatterplots could be mathematically modeled using the perceptual laws of Weber’s law [230]. Further, Chang et al. [47] also use Weber’s law to provide a guide for practitioners to select a visualization.

3.12.2 Visualization

Identifying correlation supports the decision-making process in multivariate data visualization. A comparative study of parallel coordinates, tabular visualization, and scatterplot matrix reported
tabular visualization and the scatterplot matrix have high accuracy, whereas parallel coordinates stand out in terms of completion time [75].

The Harrison et al. study compared and ranked scatterplots to other visualization methods (parallel coordinates, stacked area charts, stacked bar charts, stacked line charts, line charts, ordered line charts, radar charts, and donut charts) for correlation perception tasks and stated that Just-Noticeable Difference (JND) in correlation could be modeled by Weber’s law [121] (see Figure 3.8). Kay and Heer proposed a log-linear-based re-analysis of the results of Harrison et al.’s work [142] (see Figure 3.9). The critical finding of these works was that scatterplots stand above all other tested visualizations for precision on both positive and negative correlations.

As a counterpoint, Saket et al.’s reported that line graphs stand out for detecting correlation in terms of accuracy, time, and user preference, followed by scatterplots [244]. However, the experiments only used a small number of points, adding further complexity to the discussion of which method is truly most effective for identifying correlations. Nevertheless, the results are supported by earlier research reporting the effectiveness of line charts for trend-finding tasks [335].

3.12.2.1 Scatterplot

Correlation is one of the more extensively studied tasks in this taxonomy, particularly concerning scatterplots. One of the first perceptual studies on correlation on scatterplots by Bobkop and Karren [32] formed the basis for many other experiments. They measured a direct estimation of Pearson’s product-moment correlation coefficient, like many other correlation perception studies [57, 158, 185, 194, 217]. However, Sher et al. conducted a study about measuring the offsets of human perception of correlation when changing visual variables, and they found that humans
Figure 3.8: Ranking the effectiveness of visualizations for representing correlation [121]. Harrison et al. leveraged perceptual laws to evaluate and rank the effectiveness of visualizations for representing correlation. One interesting finding is that judgment precision had a striking variation between negatively and positively correlated data on certain visualizations, e.g., parallel coordinates. Image reproduced with permission [121].

Correlation was also among the first visualization tasks to be studied using JNDs in four experiments [77]. The study found that users become more confused with low correlation values than with high correlation values. The reason behind the effect was that the user’s judgment of discriminability increased with the increased strength in the variable association. Rensink’s perception of correlation study opened a new door to effective visualization design by demonstrating a precise way perception of correlation in a scatterplot could be modeled using Weber’s law and JND [230]. Their design studies were focused on the comparison of performance using dot size and density.
Figure 3.9: Ranking the effectiveness of visualizations for representing correlation [142]. The study by Kay and Heer presented a series of refinements to the model presented by Harrison et al. [121] (see Figure 3.8), including the incorporation of individual differences, log transformation, and Bayesian modeling. The left side shows the posterior probability distribution over the mean log (JND) for each value of $r$ using the Bayesian censored log-linear model. The right side shows the ranking and grouping of visualizations based on how precise people’s estimations of correlations are (lower JND implies higher precision). The new model demonstrated notable differences, e.g., parallel coordinates–negative and scatterplot–negative swap positions. Image reproduced with permission [142].

Best et al. investigated correlation coefficient estimation in the laboratory-based study where they found that human brain activity during correlation perception increases as correlation is decreased [30]. Findings indicated that different relationships on a scatterplot are processed differently. However, perceptually, scatterplot processing was similar, and participants used visual features to code the pattern.

A recent comparative survey included all correlation studies in a timeline taxonomy and investigated a hypothesis—viewers attend to a small number of visual features, e.g., shape, dispersion, and orientation of scatterplots to discriminate correlation in scatterplots [330]. The report compares the findings with previous pivotal correlation perception studies using Weber’s law. Scaling is another
shape property that plays an important role in identifying the correlation between the variables. An early study by Cleveland et al. reported that variables in a scatterplot look more correlated when scales are increased [57].

3.12.2.2 Map

Visualizations on maps show spatial data, and design variations are useful in identifying correlation over geo-temporal data variables. Map lineups used JNDs for evaluating correlation on choropleth maps [21]. They model the differences in visual stimuli of map color and are useful in controlling spatial auto-correlation and increase user’s confidence. The comparison of average JNDs measured across all three geographies demonstrated that JND increases and becomes more noticeable when irregularities become more regular.

The encoding choice on the geophysical maps has a direct influence on correlation judgments. A study on geo-temporal visualization investigated the task of identifying the correlation between two variables that evolve over time and space [209]. The findings showed that the design choices of geo-temporal multivariate data visualization would impact how users detect a correlation between variables over space and time. The vital design guidelines from this study are: (1) small multiple visualizations on maps are better for identifying correlation at a specific point in time; and (2) for identifying correlation for time steps on single maps, bar charts are better than other choices.

3.12.2.3 Other

Design variation, such as small multiples, overlay, and mirroring on three visualizations (bar, slope, and donut charts), were shown to influence the correlation task [201]. In the study, participants struggled to use motion animation to extract and compare the correlation between
data sets, whereas mirrored arrangements over adjacent arrangements achieved precise results. The comparison arrangement design evaluation confirms the prior hypothesis of better performance in overlaid designs versus small multiples.

The majority of the correlation studies have been conducted on scatterplots, but recent works have also diverged towards other visualization types. Studies have by-and-large shown that either scatterplots or parallel coordinates should be used for correlation tasks. However, their performance may interchange with positive or negative correlation due to the different representations of positive and negative correlations in parallel coordinates. For spatial data correlation choropleth maps can be an effective approach.

### 3.13 Compare

*Compare* is a compound task that was mentioned as an intentional “omission from the taxonomy” of Amar et al. [10]. The task of comparison often involves another subtask, e.g., retrieve a value, compute a derived value, etc., followed by comparison operation. Comparison was implicit in many of the prior tasks. For example, the find extremum task often requires comparing a set of candidate values to the rest of the data, e.g., “which cars are more fuel-efficient, Japanese cars or American cars?” [10]. This section focuses on the performance of comparison tasks based on different visual features in their design of the same visualizations to analyze the perception judgment. Based on the quantity and importance of studies of this type, I have included it as the eleventh task of the taxonomy.
3.13.1 Visualization

3.13.1.1 Bar Chart

The majority of the comparative study performed on bar charts investigate the user performance with design variations. The visual comparison task can be traced back to Cleveland and McGill’s work, where charts are shown, and different bar chart designs impact the accuracy and comparison of the viewer’s perceptual task [58]. The study reported that a comparison between adjacent bars is more accurate than between widely separated bars.

The findings from that study were extended to four different types of bar charts [282]. Some of the critical findings on bar chart comparisons are: short bars are difficult to compare; the gap between stacked bars can prevent part-to-whole comparison errors; distractors in bar effects unaligned bar comparison; and separating bars in space makes the comparison of their height more difficult. At the same time, comparing the variants of the bar chart design provides different levels of performance on completion time and standard error [272]. Variants of bar charts were compared based on the visual features in their designs, and their perception judgments were used to discover error and distraction factors. Charts with different overlays or hybrid designs that combine aspects of juxtaposition and explicit encoding with superposition are just as good or better than sole juxtaposition or explicit encoding-based charts on individual tasks. A comparison helps the designer to identify differences in the representation of data.

Other forms of design variation in bar charts are embellishments which attach aesthetics to visual display. Different bar charts with visual embellishments were compared to evaluate the accuracy, and the results demonstrate that accuracy was not worse, but at the same time, it did not provide better outcomes [264].
Comparing a visualization to a mental image is similar to statistical analysis, and thus repeated interpretation of visualization is sensitive to the multiple comparison problem—the probability of discovering false insights when visualization is examined more times or compared [336]. The study was performed on different types of bar charts to measure and compare the accuracy of user-reported insights such as shape, mean, variance, correlation, and ranking.

Another work on bar charts is compared with Microsoft Excel to show the difference in activities in terms of sequences of action and pipelines [324]. Participants tend to follow a linear pattern when using Excel, whereas while using tiles, they followed a cyclical pattern. In a study of five types of ranked list visualization, a comparison of two data items takes the longest time with the lowest accuracy on Zvinca plots [189]. A treemap outperformed all other visualizations on accuracy.

3.13.1.2 Line Chart

A series of studies have evaluated and compared the performance of line chart variants: horizon graphs\(^5\) [124, 134, 211, 227], colorfields,\([65, 110]\) braided graph [134], and small multiples [288]. A recent study assessed horizon graphs and colorfields, along with a simple line graph on the perception of similarity between time series—two patterns were considered similar irrespective of their amplitudes or their stretching along the time dimension [110]. Layered bands are more useful as chart size decreases.

In another study, where mirrored and offset horizon graphs were compared, found that estimation error and time increased at four bands in the horizon graphs [124]. At the same time, the different chart types did not affect the estimation time or accuracy. The effects of chart size of

\(^5\)Horizon graphs split the space (mainly vertically) and attempted to optimize the vertical footprint to visualize multiple time series [134].
horizon graphs and layering on comparison and estimation showed that horizon graphs performed better than line charts for small chart sizes. A time-series visualization comparison studied line charts, horizon graphs, and color fields in a similarity perception comparison where the choice of visualization affects the viewer’s perception of temporal patterns [110].

3.13.1.3 Network

Complementary views are best and increased the accuracy of network exploration tasks. Chang et al.’s work compared matrix diagrams, node-link diagrams, and weighted networks to find effective matrix representations in side-by-side views for network exploration tasks on error, completion time, and user preference [46]. Findings state node-link and matrix views are well suited for different visual tasks. Another study compared graphs either for different tasks or different datasets to measure their effectiveness. Graph edge attributes with uncertainty were visualized using two separate visual variables. For the task of comparing two graphs on overall strength or certainty showed that lightness was an effective mechanism for uncertainty [114].

Readability is another network feature evaluated in a comparison study between node-link diagrams and their matrix-based representations on generic low-level tasks [107]. They found readability depended upon graph detail, familiarity, graph meaning, and the layout used to visualize them, but the findings also reported that it deteriorated when the size of graphs and link density increased. In a recent work on the perception of graph properties, three layout algorithms—force-directed, multidimensional scaling, and circular—were modeled and compared using Weber’s law to discriminate between graphs [270]. The results showed that Weber’s law could be used to model density perception.
For the comparison task, I observed a relatively fair distribution of studies on bar charts, line charts, and scatterplots. Ultimately, the choice of visualization is more subtask-driven and design-dependent.

3.14 Discussion

I have presented a systematic review of research in the perception of visualizations. The web version of taxonomy framework is available as VisPerception: <https://usfdataviz.github.io/VisPerceptionSurvey/>. One important observation from this survey is that much of the research has been pursued through the lens of low-level tasks in terms of efficiency and effectiveness. Early works were broad in nature. For example, Cleveland and McGill’s early work demonstrated how the many different types of visual encodings influence perceptual judgments [58].

3.14.1 Visualization Effectiveness is Task-Dependent

One of the important conclusions I have seen time and again in prior work is that low-level task effectiveness varies with the dataset at hand, the visualization used, and specific design variations with the visualization. While some visualization types tended to perform better than others on average, it seems, from my observation, there is no single visualization that is suitable for all situations. Saket et al. seemed to agree in their study of five visualizations on small datasets using ten low-level tasks [244], when they stated that “No One Size Fits All.” In other words, depending on the task at hand, various visualizations perform can perform better or worse. In a similar vein, even within a single visualization type, design variations can have a serious effect on performance. Mylavarapu et al.’s study on ranked-list visualizations, which included wrapped bars, packed bars, piled bars, and Zvinca plots, quantified the differences and trade-offs for three tasks.
The effectiveness of the representations varied, as each had its own strengths and weaknesses that depended upon the task, data, and user [189].

3.14.2 Progress Understanding Graphical Perception

Investigating and evaluating the effectiveness of visualizations to optimize their visual design is a perennial topic. I witnessed the continuously evolving nature of perceptual research, with the majority focusing on visual task judgment. The recent upward trend in perception-based visualization studies, as noted in Figure 3.1 and supported by two recent related STAR reports by Borgo et al. [33] and Behrisch et al. [22], show the maturing of this area of research. Many of the works discussed in this report have applied perceptual laws to evaluate visualizations, hopefully leading to better visual design.

Much of our knowledge about perception in visualization is taken for granted, and despite the diverse set of perceptual research in visualization, many topics we have presumed as “fact” have never been sufficiently studied. Take, e.g., parallel coordinates—despite their reputation, studies have shown that parallel coordinates may not be as difficult to use as we think [263]. The point here is not entirely to discard the facts but instead to consider that re-evaluating what we know might lead to better design guidelines. This align with some of the work of Kosara [150], where the effort was to tease apart what we know and what we only think we know, using examples. The goal is to point out specific gaps in our knowledge and to assist researchers in starting investigations of the underlying theories to systematically build up a better foundation for our field.

As the research objectives and methodologies have evolved, insights have become more fine-grained and nuanced, e.g., Chung et al. observed that color saturation with size could be used as
Table 3.4: Table summarizing the number of studies I reviewed per task, visual encoding, and visualization type and reveals which area have received the most attention and those that need more work.

<table>
<thead>
<tr>
<th>Task</th>
<th>Visual Encoding</th>
<th>Visualizations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Spatial Position &amp; Shape</td>
<td>Scatterplot</td>
</tr>
<tr>
<td>Retrieve Value</td>
<td>22</td>
<td>7</td>
</tr>
<tr>
<td>Filter</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>Compute Derived Value</td>
<td>18</td>
<td>36</td>
</tr>
<tr>
<td>Find Extremum</td>
<td>15</td>
<td>14</td>
</tr>
<tr>
<td>Sort</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Determine Range</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>Characterize Distribution</td>
<td>14</td>
<td>11</td>
</tr>
<tr>
<td>Find Anomalies</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>Cluster</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>Correlate</td>
<td>14</td>
<td>10</td>
</tr>
<tr>
<td>Compare</td>
<td>24</td>
<td>22</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>94</td>
<td>84</td>
</tr>
</tbody>
</table>

an ordering variable [56]. Similarly, Szafir’s work measured perceptual judgments on color difference that varied with size encodings, e.g., bar-width, circle-radius, line-width [280]. These innovations, while enlightening, also stand somewhat in contrast to applied works, e.g., Colin Ware’s book [303] on visualization design and perception, which provides a practical view of effective design.

3.14.3 Limitations of Scope and Reproducibility

As findings become more nuanced, so too does the scope of the findings. Most individual studies come with limitations in scope, complexity, objective-goal, and sometimes with datasets or demographics. For example, studies that are performed using limited and less diversified subjects, e.g., on computer science students, need to be further investigated before their guidelines could be put into practical use for a broader population. Readers and practitioners need to pay close attention to the limitation studies, as they are critical in effectively applying their conclusions.

An integral part of understanding the limited scope of studies is considering the issue of reproducibility in studies. Reproducibility is being increasingly encouraged by communities of science to validate conclusions and to extend, re-evaluate, or specialize the original ideas [127, 219]. With a
few exceptions, e.g., [123, 142], there has not been significant consideration of replication within the perception studies I have surveyed. Several fields of science are experiencing a “replication crisis” that has negatively impacted their credibility. The problem is further exacerbated by the limited scope of visualization studies, specificity of experiments, and difficult logistics required for reproducing a study.

The lack of sufficient incentives, e.g., generating highly cited publications at high-quality venues, for reproducing studies also stands in the way. Recent suggestions for mitigating this problem revolve around tightly coupling original research with replication studies [127, 219], though this approach does not address the lack of incentives. One recent step in the right direction has been the trend of encouraging authors to make their research data available. A great set of recommendations are provided for open research practices in [297].

3.14.4 Limitations and Open Questions

Throughout this survey, I uncovered various open challenges and areas requiring more study.

1. Beyond Scatterplots, Bar Charts, and Line Charts: Many research studies in visualization are driven by popularity, familiarity, and applications. Table 3.4 provides an overview to what extent the various areas have been studied in the survey of papers. Clearly, scatterplots, bar charts, and line charts received the majority of attention in research papers. It is undeniable that bar charts and line charts are widely used to represent univariate data, while scatterplots are used to represent bivariate relationships effectively. Parallel coordinates are an interesting case—even though there has been extensive design-based research work on parallel coordinates, the number of perceptual evaluations is low. One reason may be that it is complex and
challenging to perform the perceptual evaluation. Similarly, I observe a bias in Table 3.4 to
the tasks of Retrieve Value, Compute Derived Value, and Compare. While it is unlikely that
the distribution of studies will balance out, insights gained from some of the less frequently
studied techniques can have significant value. Researchers should consider focusing their new
research efforts on the less studied tasks and visualization types.

2. Limitation of Experiment Data Collection: One topic I did not address in any detail is the
collection of human subject data used in these studies. Variations in data collection make meta
analyses difficult and can harm reproducibility by fixating on limited subject pools or specific
experimental setups. The evaluation and perceptual studies I identified most often used metrics
such as accuracy, time to completion, and subjective preference. Data collection methods range
from keyboard input to mouse input to eye-tracking (e.g., scanpath and fixation) to voice. The
experimental strategies also varied from methods of adjustment or interaction, forced-choice,
and think-aloud protocols, which can provide more cognitive and problem-solving insights.
Many of the experiments also suffer from low ecological validity, which refers to how closely the
experimental setting matches the real-world setting in which the results might be applied [170].
There is an explicit trade-off between experimental control and ecological validity [42]. Studies
with high ecological validity closely reflect real-world use, while those with low ecological
validity are often highly controlled. Finally, I found variation in the participant pools used,
generally consisting of university students or random subjects on crowdsourcing platforms. All
of these variations in metrics, apparatus, and subject pools limit the applicability studies. For
example, familiarity with the visualization, task, datasets, and design can be extremely biasing
to participants’ performance. In the same regard, other biases, such as affective priming, can
influence perceptual judgment. These limitations require serious consideration when applying conclusions to designs.

3. Relationship to Cognitive Processes: Perception and cognition are tightly linked and difficult to separate. Almost every study I discussed had some cognitive component to it. For example, estimating correlation is both perceptual and a cognitive task. In terms of the efficacy of visualization, cognition plays as important a role as perception does. For example, one of the metrics on task performance, which was largely ignored or at least largely unreported in experiments, was confidence. Based on the complexity of the task, e.g., interaction versus forced-choice, visualization type familiarity, e.g., parallel coordinates, comfort with the type or size of data, or the type of experimental study being performed, subject confidence can vary widely. There is even a possibility that the user’s confidence will vary task-by-task. A related challenge is finding the right set of subjects, which is a particular limitation when recruiting participants through crowdsourcing. Due to the population diversity, their experience and confidence vary, and it is difficult to check quality. It fits then for researchers to evaluate the participants’ level of confidence in task judgment, and accordingly, data quality should be evaluated. New studies should consider the participant’s level of confidence and how this affects high- and low-level tasks.

3.14.5 Conclusion

I have presented this report with a particular focus on the links between visualization types, visual encodings, and tasks. Throughout taxonomy, I wanted to emphasize perception-based research findings and their impact on visual encoding and visualization choices. I believe this report will be
a valuable starting point for those designing visualizations and researchers looking to advance the state-of-the-art in perception-based visualization research. It should be noted, however, that the aim of this survey was not to directly summarize the visualization recommendations but instead to provide a broad understanding of the topics that have been evaluated. Furthermore, all of the perceptual studies were studied under limited conditions and come with caveats that I could not enumerate.
Chapter 4: You Can’t Publish Replication Studies (and How to, Anyway)

“Replication: Attempt to confirm, expand, or generalize an earlier study’s findings [127]”

4.1 Introduction

A replication study is a re-evaluation, re-confirmation, or extension of an original study. These studies can be performed under similar experimental conditions to the original studies to validate conclusions or under varying experimental conditions to gain more knowledge [127, 210], and they provide an important concrete baseline, which is useful to improving cross-study validity [157]. The journal *Science* labeled replication as the “scientific gold standard” [133] and in others’ words “in replication, the private chimera becomes the communal fact” [19].

Unfortunately, the positivist notion of reproducibility does not offer the novelty of qualitative research itself [279]. In a literature review of 891 papers by Hornbaek et al., they found only 3% of papers in Human-Computer Interaction (HCI) attempted replication [127], and at a recent BELIV workshop multiple researchers pointed out that it is rare to find replication work in information visualization research [151, 278].

Efforts have been made to encourage researchers to improve reproducibility and conduct more replication studies. For example, it is increasingly common for authors to voluntarily (and sometimes mandatorily) provide their experimental data and other supplemental materials to accommodate

---

6This chapter was published in In Proceedings of VIS’19: IEEE Conference on Visualization. Workshop on Vis X Vision, 2019. Permission included in Appendix B.
future replication. Since 2012, the EuroRVVV [98] Workshop on Reproducibility, Verification, and Validation in Visualization has been promoting and supporting replication in visualization. The RepliCHI workshop, organized as a part of the ACM CHI conference, promotes a shift towards favoring replication in the HCI research community [318]. Finally, the BELIV [23] workshop at IEEE VIS conference is also focused on replication and reproducibility [151, 278, 293].

A highly debated usability evaluation paper in ACM CHI highlighted the lack of replication studies in HCI [112]. It argued that reviewers do not value replication work, and therefore papers are not published, despite the intrinsic value in their work. The reluctance from the reviewers and publication venues to consider replication studies as a valuable contribution has lead to few replication studies being published [112, 127, 318]. Most of the replications studies successfully published are of controversial findings or of highly-cited works [39, 58, 81, 112, 127, 139, 238]. Instead of attempting replication studies, HCI and information visualization researchers have been pushed towards novel and organic findings [127, 193, 318]. Nevertheless, savvy researchers have been successful at publishing replication studies in less conventional ways, in particular, by embedding them into other sufficiently novel studies. Many of these studies have been at the intersection of vision science and visualization.

Recent workshops and conferences on vision science and visualization have established the potential for methodologies, experimental techniques, and user studies on perceptual judgements from vision science to benefit visualization. However, cross-field contribution is challenging. For vision scientists wanting to interact with the visualization community, replication can be a viable and relatively low-risk area where they can contribute.
In this position paper, I highlight how researchers have integrated replications of prior studies into new studies by including them with cutting-edge contributions. In this way, these new works confirm the prior studies and introduce a novel idea or methodology. In particular, I found three common methods of integration: (1) re-evaluating under different demographics and/or participant environments; (2) expanding upon a study’s conclusions by new experiments that elucidate additional information or deepen understanding; or (3) specializing the knowledge to a specific domain by elaborating on experimental conclusions. To demonstrate these three methods at the intersection of vision science and visualization, I highlight a number of replicated works of Cleveland and McGill’s graphical perception paper [58], whose objectives are similar to the original ranking of quantitative judgment effectiveness of graphical encodings. I review their innovative contributions and contribution to replication-based validation. Finally, I discuss our perspective on how vision scientists can use this information for producing replication works in the current publication environment.

4.2 Background

The visualization field is “an empire built on sand” with a weak foundation and in need of replication to strengthen many of the assumptions used by the field [150]. In a recent study, Kosara and Haroz pointed out the replication crisis—very few replication studies are attempted in visualization, let alone published [151]. They examined six threats to the validity of studies in visualization and provided suggestions for replications, like outlining study design flaws and understanding or re-running misinterpreted results. Their suggestions help to minimize the threats to producing scientifically sound work.
Sukumar et al. provided guidelines for experimental design to encourage researchers to conduct more replication studies in information visualization [278]. They provided a list of possible experimenter biases that can occur related to devising hypotheses, independent and dependent variables, tasks, experimental procedures, sampling, experimenter behavior, and experimental setting, and they focused their discussion on designing and running sound experiments.

Hornbaek et al. developed a prescriptive definition for replication studies [127]. They stated that a replication must name and reference the original work, and they must state how their work confirms or extends the prior study. They further distinguished the replicated work with three categories—strict, partial, and conceptual—based on their literature review. A similar type of distinguished category can be found in Kosara and Haroz work—reanalysis, direct replication, conceptual replication [151]. These characterizations vary by the amount of originality from previous work that has been kept intact. Strict replication uses the same variables with the intent to reproduce the exact same study. Strict replications usually only replicate and confirm the original findings, which is what draws reviewers reluctance. Partial replication modifies the original study for testing within a different environment or with different participants. Conceptual replication studies investigate the same study but with different metrics, settings, or judgment criteria.

I distinguish our contribution from previous studies [127, 151, 278] by focusing on a taxonomy that does not focus on the quantity of overlap or similarity of study design in replication studies, but instead it focuses on the types of novel contributions that are associated with the replication studies.
4.3 Novelty and Replication: Taxonomy

Hornbaek et al.'s work [127], along with the Kosara and Haroz work [151], essentially considered the level of similarity to the original study when classifying replication. As a complementary measure to that, I consider classifying replication studies by the *objective* of the novel contributions of the study. By understanding the types of novel contributions that blend with replication studies, other researchers, like those from vision science, can mimic the contribution styles to produce their own replication studies.

Our evaluation of prior work shows that the vast majority of replicated work in information visualization falls within one of the three following categories.

4.3.1 Re-evaluate an Experiment’s Objective

As practices change, software and hardware advance, and new techniques or information become available, some research studies whose conclusions were once considered solid require re-evaluation using these new contexts. Re-evaluation studies confirm the findings of an original study with different environment setting, while attempting to reproduce the objectives as closely as possible [127]. These replications help to establish if results from the prior study can be repeated to increase confidence in its validity. For example, Kosara and Ziemkiewicz replicated [153] their own earlier studies [26, 265] on pie charts using Amazon Mechanical Turk in order to re-evaluate whether an online environment produced the same results. Additional examples of this type of replication study are [123, 142].
4.3.2 Expand an Experiment's Objective

Due to the efforts required for performing human studies, the scope of studies is often kept small, leaving the need to expand conclusions with followup studies. Replication studies can be expanded beyond the objectives and conclusions of the prior studies by conducting themselves under different experimental conditions to make a novel contribution. By experimenting under different conditions, these replication studies serve as alternative means to validate prior results or as a means to generalize results [121, 131, 148]. For example, Rensink and Baldridge investigated the perception of correlation in scatterplots and suggested that Weber’s law was useful for modeling it [230]. A replication study by Harrison et al. supplemented new conditions, in order to broaden the scope into a number of additional visualization types [121] (which was itself extended further through re-evaluation type paper [142]). Some additional examples of these types of replication studies are [146, 203, 226].

4.3.3 Specialize an Experiment's Objective

Studies often result in generalizable conclusions that need to be studied in new or more specific contexts than those of the original study. By taking the original study as the fundamental base, these works consider if and how much of the knowledge acquired in the original study is transferable to a new, different, or more specific domain. Performing these replication studies of different contexts under similar conditions with respect to setting, experiment environments, or metrics, aims to establish the validity within that specialized domain. Rensink and Baldridge’s [230] study on modeling correlation perception was also replicated by Yang et al. [330] in order to further investigate visual features in modeling perceptual processes. The objective of finding perceived
correlation in a scatterplot is synonymous with perceiving its visual features and quite unrelated to one’s statistical training. The results of this replication study showed how visual features provide a baseline for model-approaches in visualization evaluation and design. Additional examples of this type of replication study can be found in [124, 280].

4.4 Case Study

With a focus on perceptual work in information visualization, I selected the seminal and highly cited work by Cleveland and McGill on graphical perception [58] to highlight examples of replication studies. Graphical perception is an anchor of visualization research [303] with the potential to improve the efficiency and effectiveness of the automatic representation of data [177]. I have observed the vast influence of this paper in last two decades of information visualization
research. According to Google Scholar, this work has been cited more than 2040 times as of October 8, 2021, and aspects of their study have been replicated many times.

![Graphical encoding channels](image)

**Figure 4.1:** Reproduction of the Cleveland and McGill’s graphical encoding channels [58].

### 4.4.1 Graphical Perception

Research has demonstrated viewer’s perceptual judgment significantly influences effective visualization design. The representation of data in a visualization is encoded with specific elements on the display, also known as the graphical encodings, include *position*, *length*, *angle*, *area*, *volume*, *shading*, *direction*, *curvature*, and *color* [213]. Figure 4.1 represents these 10 elementary *perceptual tasks*\(^7\) from Cleveland and McGill’s work that people use to extract quantitative information from graphs.

Understanding the role of perception in the choice of graphical encodings is critical to visualization designers. Based on 10 common graph types—distribution function plots, bar charts, pie charts, divided bar charts, statistical map, curve-difference charts, Cartesian graphs, triple

---

\(^7\)The term *perceptual task* originates from the concept that viewer performs a mental task to extract quantitative values represented on graphs [58].
scatterplots, volume charts, and juxtaposed Cartesian graphs—Cleveland and McGill ranked these perceptual tasks by the accuracy of quantitative information extraction. Mackinlay produced one of the earliest comprehensive rankings of graphical encodings by data type [177]. The ranking has been further validated and elucidated through followup (replication) studies [71, 111, 123, 124, 191, 226, 246, 256, 280].

4.4.2 Example Replication Studies

I surveyed major visualization publication venues (IEEE TVCG, ACM CHI, EG EuroVis, and IEEE PacificVis) for papers that cited and then replicated aspects of Cleveland and McGill’s work. Our non-comprehensive analysis is primarily limited to the eight papers listed in Table 4.1. For each, I provide the objective of the replication study, followed by a high-level description of the novelty added to the replication, which contributed to the visualization research.

4.4.2.1 Re-evaluate an Experiment’s Objective

Heer and Bostock’s graphical perception study replicated ranking on effectiveness of visual encodings, such as length, position, and angle, using an alternative crowdsourced subject pool on Amazon Mechanical Turk (AMT) [123]. Further, they extended the study on additional encodings, such as circular area (e.g., bubble charts) and rectangular area (e.g., treemaps). The ranking order that resulted from their studies were similar to Cleveland and McGill’s rankings. The main novelty of this chapter was not to validate the findings of Cleveland and McGill, but to test the viability of online user study like crowdsourcing. The results showed that AMT could serve as a viable user study platform for visualization research.
Another replication of Cleveland and McGill was carried out by Saket et al. on 12 graphical encodings to study their effectiveness in terms of task completion time and accuracy, when using them for interaction [246]. The objective of replication was same as that of original study but re-evaluated on interactive graphical encodings. Their ranking followed and confirmed the findings of Cleveland and McGill’s, except for a significant difference between length and angle in terms of accuracy.

4.4.2.2 Expand an Experiment’s Objective

Hullman et al. [130] and Harrison et al. [120] replicated the study of Cleveland and McGill but ranked their quantitative judgement effectiveness on the basis of social information\(^8\) and affective priming\(^9\), respectively. The AMT-based perceptual studies demonstrated social information and affective priming can significantly influence user’s visual judgment [120, 130]. The findings on social information can be applied to collaborative visualization systems to produce more accurate results on individual interpretations in a social context, and the findings on affective priming showed that it can influence accuracy in common graphical perception tasks.

The concept of graphical perception has extended to various branches of visualization—maps, color, visual properties, etc. Another replication of Cleveland and McGill, towards color mapping on continuous maps, found that spatial frequency significantly impacts the effectiveness of color encodings [226]. The granular level of novelty to this work is based on conceptual replication of previous work and applying it to a new domain of continuous maps.

---

\(^8\)Social information represents the creation and processing of information from multiple features by a group.

\(^9\)Affective priming is the impact of emotional biases, whose study involves manipulating valence and/or arousal via emotional stimuli.
A partial replication of Cleveland and McGill studied how visual encodings, such as color, shape and size, affect a user’s way of interpreting data [71]. A perceptual kernel\textsuperscript{10} is estimated for set of perceptual stimuli, based on size, shape, color, and combinations, to assess the effectiveness of visual representations from reported results of crowdsourced experiments. They compared six stimuli using different set of judgment types—Likert rating among pairs, ordinal triplet comparisons, and manual spatial arrangements—to existing perceptual kernels and demonstrated how kernels can be applied to automate visualization design decisions. The novel contribution from this replication work is fixed on similarity judgement using perceptual kernels, extending the concept of the prior work to a particular domain.

4.4.2.3 Specialize an Experiment’s Objective

Talbot et al.’s [282] study on bar charts provides insights on comparing adjacent, separated, aligned, and non-aligned bars in types of bar charts. This replication used the foundational work of Cleveland and McGill to focus on perceptual tasks related to bar charts only. Distractors (i.e., intermediate bars between bars being compared) affect the comparison between types of bar charts, and inconsistent placement of marking dots in the original study affect user accuracy on perceptual tasks.

In similar fashion, the specialized domain replication work of Heer et al. [124] extended the concept of graphical encoding effect of user judgement to the specialized area of chart height and layering on time and accuracy of a value comparison task. Their findings on estimation accuracy

\textsuperscript{10}Perceptual kernel is a distance matrix derived from aggregate perceptual judgments. It contains pairwise perceptual dissimilarity values for a specific set of perceptual stimuli.
across charts identified transition points in smaller charts, where accuracy and estimation time decreases with size.

For all of the studies mentioned, the additional novelty of the papers helped it stand out beyond just the replication—the studies verified previous results, but they also covered a larger parameter space and/or came to different conclusions than the original study.

4.5 Discussion

The overwhelming value of reproducibility is undeniable—from the ability to have third party verification of claims and conclusions, to the development of new insights expanding upon old conclusions, to understanding changes in user expectations and technologies with respect to prior study findings. While general effort toward improved reproducibility have had some success (slow progress but trending in the right direction), replication studies have remained somewhat in the shadows. Further attention needs to be paid to this particular area of reproducibility. One natural avenue is for vision scientists interested in applying their expertise to the area of visualization to contribute replication studies that either re-evaluate, expand, and/or specialize previous studies.

4.5.1 You Can’t Publish (Strict) Replication Studies

As long as novelty remains a necessary contribution that reviewers in the visualization community acknowledge, the simple fact is that it will be incredibly difficult to publish strict replication studies. By their nature, the contribution of strict replication studies is not novelty, thus the reluctance from the reviewers to acknowledge any value. Though some may exist, I did not find any strict replication studies of Cleveland and McGill during our literature survey. Our opinion is
that it remains in the best interests of researchers, in vision science or otherwise, to avoid trying to publish any strict replication study.

In many ways, the struggles of replication parallel those of application papers in visualization. When application papers are discussed in the halls of the IEEE VIS conference, everyone agrees on their value and wishes there was more acceptance for them. As soon as those individuals review an application paper, a stamp of ‘limited novelty’ (the review equivalent of the ‘kiss of death’) is applied and the paper is promptly rejected. It is only by wrapping the application in novelty that these papers are ever accepted in the main conference tracks. A variety of attempts have been made to correct for this issue, most have been failures. The introduction of Application Spotlights at IEEE VIS conference, 2019 was the latest attempt, whose success has yet to be determined. The point is, those looking to promote reproducibility and replication studies should look to the history of application papers for some insights as to what approaches are likely (and unlikely) to succeed moving forward.

4.5.2 How to Publish Replication Studies Anyways

In this chapter, I argue that the best way for individual researchers to publish replication studies is to distinguish the work with the help of added novelty. Considering partial and conceptual style replications bridges the gap between original work and the innovations required for publications. For those new innovations, I demonstrated how prior works used re-evaluation, extension, and specialization to help frame their novel contribution around the replication study, enabling the replication to provide value to the overall contribution of the paper.
For vision researchers new to visualization, the concept of replication can serve as a bridge into the field. Replication enables the researcher to become deeply familiar with a specific visualization topic, while contributing to the field. Taking the replication as a base, re-evaluating, expanding, or specializing enables them to contribute added novelty to the field. In this way, the familiarization process (i.e., the replication study) has both personal and community value. For example, re-evaluation work can be used as a foundation for a researcher introducing themselves to visualization community, or an expansion can be used to evaluate conclusions under different conditions, not previously considered. However, I believe specializing represents the best opportunity. Vision scientist can leverage their in deep knowledge of human perception in efforts to replicate and specialize prior visualization studies, thereby bringing innovative ideas to the visualization community. In similar fashion, findings from the vision science community can be replicated and specialized into the context of visualization. In the end, visualization replication studies by vision scientists represent a win-win. First, it engages both communities in a dialog that advances knowledge in both communities. Second, we all benefit when the experimental conclusions we rely on are re-validated and further elaborated upon.

Replication does not necessarily need new experimentation. Already completed studies can be utilized to generate new state-of-the-art work. Researchers can make use of available experimental data, study designs, and comparable results from original studies in order to formulate new high-level research questions. A re-evaluation of an earlier study can reduce the possibility that conclusions are the result of a statistical fluke, flawed analysis, or a flaw in the study design [151]. New perspective on the analysis of that data can be used to expand or specialize the domain of the work.
Finally, it is important to recognize that many non-replication studies over the years could have corroborated the conclusions of earlier studies by performing small replication studies of key findings. This chapter does not necessarily advocating that all or even the majority of prior studies should be replicated. However, reviewers could encourage more replication by allocating “bonus points” in reviews containing some form of replication. The research community is already seeing this with reproducibility in general. In a personal communication with IEEE VIS’2018 InfoVis chair, they noted a measurably higher average score for papers that included their data in the submission.
Chapter 5: Modeling Cluster Perception in Scatterplots

5.1 Introduction

Scatterplots are commonly used to reveal several types of relationships between quantitative variables [104]. Numerous perceptual studies have evaluated the effectiveness of scatterplots in low-level tasks that include assessing trends [77, 196], measuring correlation [30, 121, 230], and average and relative mean judgments [109]. Clustering, in particular, is an aggregate-level task [166, 188, 247] that has been utilized in a variety of applications, e.g., weather forecasting, text analysis, and large-scale data analysis [165, 281, 304, 319]. Clustering occurs when patterns in the data form distinct groups [10, 248]. However, at its core, clustering is an ill-posed problem, as the “correct” clustering depends upon multiple factors [78, 101] 11.

When considering clustering in scatterplots, several factors play a role in how they are perceived. Data aspects, such as the data distribution size/type and the number of data points, can influence the visual presentation. On the other hand, visual encoding properties, such as mark type, size, and opacity, have the potential to influence perceptual judgments [58]. What is not well understood is how these various factors, many of which are under the control of the visualization designer, influence the perception of clusters. The presentation of data is particularly important when considering that a biased representation of the data may provide an inaccurate summary, leading to invalid conclusions [111, 147, 163, 280].

11This chapter was published in IEEE Transactions on Visualization and Computer Graphics (Volume: 27, Issue:2, Feb. 2021), DOI: 10.1109/TVCG.2020.3030365. Permission included in Appendix B.
In this chapter, I explore the multi-factor judgments used in identifying clusters in scatterplots through a crowdsourced user study. Based upon this study, I develop two models for the perception of clusters in scatterplots, using a data structure from Topological Data Analysis, called the *merge tree* [305]. I validate the models on a variety of variables—*the number of points, cluster distribution size, size of data points, and opacity of data points*—to verify the accuracy of the models and analyze their effects. Our results show that the perception of the number of clusters does indeed depend upon all 4 factors. Moreover, I show that the merge tree-based models do match an average user’s perception of the clusters in a given scatterplot.

Finally, I demonstrate how the models can be used to optimize visualization designs. While some variables, such as distribution size, are difficult to control in a visualization, designers can use this models and findings as a guideline to balance the design factors that they do have control over—the number of points shown (e.g., via subsampling [50, 128]), data point size [281, 290], or opacity [180, 186]—to optimize the saliency of the clusters in a visualization. A demo of the approach can be viewed at <https://usfdatavisualization.github.io/TopoClusterPerceptionDemo>.

I provide brief coverage of clustering in scatterplots and perception of the visual factors evaluated in our study.

5.1.1 Clustering in Scatterplots

Clustering plays an important role in exploring and understanding many types of data [247, 248]. A design factor survey defined clustering as a high-level data characterization—the ability to identify groups of similar items [248]. Amar et al. presented a set of tasks for visual analytics that defined clusters as having “similar attribute values in a given set of data” [10].
5.1.1.1 Taxonomies of Clustering Factors

Identifying clusters is directly influenced by the perception of cluster separation, and much of our understanding has come from studying dimension reduction (DR) techniques. Lewis et al. compared the effectiveness of DR techniques using human judgments and concluded that T-SNE performs better than other commonly used methods when expecting clusters in the data [161]. Etemadpour et al. showed, however, the performance of DR techniques also depends on data characteristics [96], e.g., the separability of clusters, and later created a user-centric taxonomy of visual tasks related to clustering in DR techniques [95]. A taxonomy of visual cluster separation in scatterplots used a qualitative evaluation to identify 4 important factors—scale, point distance, shape, and position [256]. The taxonomy gives a context to our visual factor selection. Sedlmair and Aupetit later evaluated 15 class separation measures for assessing the quality of DR using human input for building a machine learning framework [254] and later extended the framework to include an even greater number of measures [15].

5.1.1.2 Clustering in Non-Scatterplot Contexts

Clustering has been studied in other types of visualization, such as text [7], maps [165, 304], and bubble charts [281]. A task-based evaluation found that on small data, bar and pie charts outperformed tables, scatterplots, and line charts in clustering tasks [244]. The performance in cluster perception in pie charts is traced back to its effectiveness in facilitating proportional judgments through a part-whole relationship [87, 271]. Similarly, I hypothesize that the relative distance between clusters and the relative density of the image influence cluster identification.
5.1.2 Factor Selection on Scatterplots

From this collection of possible factors as discussed in the section 2.2.1, I focus this study specifically on the factors that most influence visual density, including the distribution of and distance between concentrations of points, the number and size of data points, and data point opacity in the visualization.

5.1.2.1 Point Distribution

Several prior studies have investigated the influence of the distribution of data points on cluster perception. An early study of 8 participants on 24 homogeneous dot patterns studied the impact of varying densities and gaps between 2 square-shaped clusters [200]. Sadahiro later developed a mathematical model to represent cluster perception in point distributions based on 3 factors—proximity, concentration, and density change—and suggested perception is significantly influenced by the concentration and density change [242]. Similarly, the scagnostics density property identifies concentrations of points directly influenced by the distribution of points [317].

5.1.2.2 Number of Data Points

Sadahiro also showed that the higher the number of points in a given area, the higher the chances are that they would be perceived as a cluster, due to increased density [242]. Gleicher et al.’s empirical study asked participants to compare and identify average values in multi-class scatterplots [109]. It demonstrated that judgments are improved with a higher number of points.
5.1.2.3 Size of Data Points

The size of symbols is an important factor in visual aggregation tasks in scatterplots [281]. As the size of data points increases, so does the density, which directly influences cluster perception [242]. Symbol size also has a direct influence on discriminability in certain tasks [163], e.g., in color perception tasks [275]. Szafir’s study on color-difference perception found that perceived color difference varies by the size of marks [280]. Size also influences search task effectiveness. Gramazio et al.’s study on target search demonstrated that while the quantity of data points has little effect on searching for a target, increasing symbol size reduces search time in a display of random points [111].

5.1.2.4 Opacity of Data Points

As the number of data points increases, scatterplots suffer from overplotting, which obscures the data distribution. Reducing mark opacity can alleviate overplotting to aid various visual analytics tasks [247], e.g., spike detection in dot plots [67]. Furthermore, different opacity levels aid in different visual tasks—while low opacity benefits density estimation for large data, it also makes locating outliers more difficult [186]. Matejka et al. defined an opacity scaling model for scatterplots that is based on the data distribution and crowdsourced responses to opacity scaling tasks [180]. Still, their study did not evaluate how a scatterplot design based on data symbol opacity can affect user performance on visual analysis tasks. Somewhat related to opacity is luminance, which can be modeled using extreme end lightness [164], creating a popout effect [115].
5.2 Study Methodology

I investigate how visual factors affect subject responses in the task of counting the number of clusters in a scatterplot. From this, I build and analyze two models to estimate the number of clusters an average user would perceive. One model is based on the separation distance between distributions, and the other uses the visual density of points.

5.2.1 Factors

Data are presented as point marks (i.e., circles ⬤) on the scatterplots and groups of similar objects form clusters. Based on our review of prior work, I chose to use a normal distribution to generate clusters, and I selected the following experimental factors—Distribution size (S), Number of data points (N), Size of data points (P), Data point opacity (O)(Figure 5.1):

![Selected factors for experiment and user study.](image)
5.2.2 Experiments Setup

I designed our experimental study in 3 stages: (1) a preliminary experiment to calibrate the experimental factors; (2) a crowdsourced Amazon’s Mechanical Turk (AMT) experiment to validate our models; and (3) a follow-up AMT study to elaborate upon 1 of our models.

5.2.3 Data Generation

Datasets are synthesized using 5 input parameters (Figure 5.2): stimuli dimensions ([X × Y] pixels); number of clusters (C); distribution size, i.e., standard deviation (S pixels); number of points (N); and signal-to-noise ratio (SNR). First, C cluster centers are randomly placed within a “safe zone” defined as 1 standard deviation from the stimuli (image) border, in other words, \( x \in [S, X - S] \) and \( y \in [S, Y - S] \). Each cluster is assigned an equal share of the available points \((N/C)\). Points are randomly placed around their cluster center using a normal distribution with a standard deviation of S pixels. Points outside of the image dimensions are discarded without replacement. Next, an additional \( N/SNR \) points representing noise are placed randomly using a uniform distribution across the image dimensions. Finally, to generate images, 2 more input parameters are used: point size (P pixels) and point opacity (O). The points are drawn as filled circles of P area with O opacity. Example stimuli are shown in Figure 5.3.

In all experiments some inputs were kept constant:

- Stimuli dimensions ([X × Y]): [550px × 550px] — The vertical size was selected such that the image would fit on the majority of desktop monitors without scrolling [274]. The horizontal resolution was selected to match, avoiding any directional bias.
Figure 5.2: Illustration of the data generation for user study. All plots are $550 \times 550$. Cluster centers are placed within a “safe zone”, 1 standard deviation away from the boundary. Points are sampled from a normal distribution (blue), and points outside of the image are discarded without replacement (gray).

- Signal-to-noise ratio ($SNR$): $10 : 1$ — I manually optimized the $SNR$ by looking for a high level of noise that would not overwhelm the clusters. I ended at $10 : 1$, making the maximum total number of data points in any given dataset $N + 0.1 \cdot N$.

5.3 Preliminary Experiment

I performed a preliminary user study to test initial hypotheses and calibrate parameters for the larger AMT experiment. Based on our observation and study of prior work, I drafted the following hypotheses:

[H1] The distribution size of clusters affects the accuracy in cluster count identification in scatterplots.
Figure 5.3: Example stimuli with the same cluster centers but varying factors. Stimulus have same cluster centers ($C = 5$), but varying distribution size ($S$), number of points ($N$), and size of points ($P$).

[H2] The number of data points affects the accuracy in cluster count identification in scatter-plots.

5.3.1 Properties and Data Generation

As aligned with previous empirical studies, e.g., [244], I selected parameter values to maintain a reasonable level of difficulty. I designed the task such that the response time for a single stimulus would be 5 to 20 seconds. I selected the following experimental factors:

- Number of clusters ($C$): $\{4 - 12\}$ — The number of clusters was selected using trial-and-error to avoid tasks that were too easy (i.e., trivial to count) or too difficult (i.e., larger number or sparse clusters).
• Data point size/area \((P)\): \{20_{px}\} — Experimental calibration was not needed for point size, as reasonable values could be determined analytically. Therefore, the point size was fixed in order to calibrate other factors.

• Number of data points \((N)\): \{1000, 5000, 10000\}

• Distribution size \((S)\): \{20_{px}, 35_{px}, 50_{px}, 65_{px}, 90_{px}\} — \(N\) and \(S\) were the main factors to test/calibrate. The value ranges were selected using our observation of sample stimuli and judgment of factors from prior work, considering sufficient range, minimum and maximum values, and the number of experimental conditions that could be reasonably tested.

• Data point opacity \((O)\): \{100\%\} — Points were fully opaque.

The dependent variable I tested was:

• User-selected number of clusters \((U)\): \([1 – 15]\)

Dataset generation for the preliminary experiment was done in the following manner—for every combination of \(S\) and \(N\), 500 stimuli (i.e., images) were generated with a random number of clusters, \(C\). Other parameters were fixed as described, leading to a pool of \(|S| \times |N| \times 500 = 7500\) stimuli.

5.3.2 Study Procedure

I developed a webpage for the experiments, where each participant was shown 50 images from the pool of 7500, one at a time, and asked the number of clusters they could see. Answers were recorded using a drop-down box with options \(1 – 15\). The maximum allocated time for each task was 20 seconds. At the expiration of time, the page was automatically advanced. To mitigate any effects
or bias, I placed a blank screen between every 2 tasks [122]. At the beginning of the experiment, I included a brief introduction to clustering and 3 training tasks for each participant, which were similar to the study tasks that followed. The experiment was expected to last 20-30 minutes, including demographic details and training tasks.

I recruited 30 participants from the College of Engineering at the University of South Florida for the IRB approved study. Participant ages ranged from 18-28 ($\mu_{\text{age}}=23$), with 24 males and 6 females. No compensation was provided. In total, 50 trials $\times$ 30 participants $= 1500$ responses were collected. While performing data quality checks on the responses, I found discrepancies—participants responding to stimuli in less than 1 second or those with responses of 1 cluster to all stimuli—in 4 participants results and removed them from analysis, leaving 1300 responses ($26 \times 50$). I further identified and removed 161 stimuli that had been reused from the pool only keeping the first occurrence$^{12}$, leaving 1139 responses for analysis.

5.3.3 Analysis and Result

![Histogram of user responses against frequency](image)

Figure 5.4: The histogram of user responses against frequency. The histogram plots differential, $D$, against frequency for the preliminary experiment, appears as a truncated normal distribution.

$^{12}$I acknowledged this is a flaw in our preliminary study design. However, since our primary goal was parameter calibration, the experiment still has value. I avoid this bias in our AMT experiment by generating stimuli per participant.
To measure accuracy for a given scatterplot, $\tau$, I use the differential: $D(\tau) = U_\tau - C_\tau$, where $U_\tau$ is the user response and $C_\tau$ is the number of clusters in the data. I analyzed the differential against the independent factors *distribution size* ($S$) and the *number of data points* ($N$) using a 2-way ANOVA test. I also calculated partial eta-squared ($\eta^2$). I observed that $S$ and $N$ have a significant effect on the accuracy in identifying the number of clusters, $(F_S(4, 1130) = 48.57, p < 0.01, \eta^2 = 0.12)$ and $(F_N(2, 1130) = 8.29, p < 0.01, \eta^2 = 0.02)$, respectively. These results confirm [H1] and [H2].

Although I found a significant effect, user accuracy had a low average $\mu_D = -3.27$ and a high standard deviation $\sigma_D = 2.66$ (accurate predictions would have an average of 0 with a small standard deviation). Figure 5.4 shows the histogram of differentials, which appear as a truncated normal distribution. The negative shift in the $\mu_D$ revealed that of the *number of points* or *distribution size* alone is insufficient to model the number of clusters users would perceive—an accurate model needs to consider the overlap of clusters. For example, in Figure 5.3, all images have an identical number of generated clusters, but the interaction between clusters causes differing numbers of clusters to appear. Instead, the distance between clusters or the visual density of the data needs to be considered as an additional factor in cluster perception modeling. The next section introduces models that each considers one of these factors.

### 5.4 Topology-based Modeling of Clustering

I propose 2 models for capturing human perception of clusters based upon approaches from Topological Data Analysis (TDA) [305]. TDA is a set of approaches used to study the “shape” of data, including scalar fields [235, 284], vector fields [299], and high-dimensional data [55, 179].
Both models capture the clustering structure using a data structure called the *merge tree*. The merge tree encodes a series of topological events in the form of creation and merging of components (specifically, 0-dimensional homology groups), based upon properties of the space under a real-valued function. The first model, based upon the distance between cluster centers, is captured using a technique called *persistent homology* [85]. The second model, based upon the visual density of points, is captured by calculating the join tree of a scalar field [44].

5.4.1 Distance-based Model

The distance-based model tries to capture human perception of clusters by considering the spatial resolution at which 2 or more cluster distributions will blend to be perceived as 1. I did this using the technique of persistent homology (PH) [85]. I provide a simplified view of PH under our limited context. For a detailed introduction, see [86].

Construction begins with a finite set of points \( V \) representing cluster centers embedded in Euclidean space (i.e., their positions on the scatterplot). Given a real number \( D \geq 0 \), I consider a set of balls of diameter \( D \) centered at points in \( V \). Continuously increasing the diameter, \( 0 = D_0 \leq D_1 \leq D_2 \leq \cdots \leq D_m = \infty \), forms a 1-parameter family of nested unions of balls. If at a given diameter \( D_i \), 2 balls overlap, I consider these balls as a single component. Figure 5.5(b) shows an example dataset with 4 values of \( D_i \). As \( D_i \) increases, more balls intersect and merge into larger components. At \( D_\infty \), all balls will overlap, forming a single component.

To compute the PH, the points \( V \) form the vertices of a graph. A 1-simplex (an edge) is formed between 2 points in \( V \) if and only if their balls intersect (i.e., the distance between them is \( \leq D_i \)). Sweeping \( D_i \) from 0 \( \rightarrow \infty \), as \( D_i \) increases, new edges are added to the graph. Components
are efficiently calculated at each step by finding connected components of the graph using the set
union data structure. The total computation time is $O(|E|\alpha(|V|))$, where $E$ are the edges of the
graph, and $\alpha$ refers to the inverse Ackermann function, an extremely slow-growing function. At $D_{\infty}$,
PH forms the complete graph. Therefore, there are $O(|V|^2)$ edges.

Figure 5.5: Distance-based model representation. With the distance-based model, (a) given the
input data, (b) cluster centers are analyzed at a series of ball diameters. Connected components
receive a unique color at each diameter. (c) The scan of diameters leads to the creation of the merge
tree, which marks topological events (i.e., creation and merging of components) with nodes.

Creating the merge tree from the prior construction is relatively simple. The merge tree is
parameterized with respect to $D$. At $D_0 = 0$, all cluster center components are born. In other words,
the balls have 0 volume. These birth events appear in the merge tree as 1 node per cluster, e.g.,
see the bottom of Figure 5.5(c). As $D_i$ increases, when 2 components first merge at a given $D_i$, a
merge node is added to the merge tree at $D_i$ connecting those components. For example, at $D_1$, the
purple and pink components intersect, causing them to merge into a single component. From that

112
point forward, 1 of the merged components dies (i.e., no longer exists), while the other takes on the identity of the new merged component (in this context, it does not matter which). Referring back to Figure 5.5(c), when purple and pink merge at $D_1$, pink dies, while purple takes on the identity of the merged component. When the components finally merge into a single component, yellow in our example, this component dies at $\infty$. In other words, no matter how large the balls get, that 1 component will exist. It is also relevant to note that this particular construction has many parallels to single-linkage hierarchical clustering.

This model has 2 main limitations: (1) it assumes that clusters are isotropic and have similar distributions; and (2) it requires knowledge about the location of cluster centers. Our next model uses a related framework to overcome these limitations.

5.4.2 Density-based Model

The density-based model attempts to directly identify the relative visual density at which users will differentiate between clusters. The density-based model is found by calculating the join tree of a scalar field. I again provide a simplified treatment—for a detailed description, see [44].

First, a 2D histogram of the visual density is created for the scatterplot (i.e., a density plot). The image plane is divided into a set of grid cells of uniform width and height (selection of this resolution is discussed in our evaluation). Within each grid cell, the number of white pixels is counted, and this is considered the density$^{13}$, $f_{xy}$. For illustrative purposes, this value is mapped to the range $F \in [0, 255]$, where 0 is empty (i.e., completely black) and 255 is full (i.e., completely white), as shown in Figure 5.6(a).

$^{13}$I acknowledge this is not the usual calculation of density, e.g., see [69], which would count the number of black pixels. However, our configuration makes the remainder of the discussion easier.
Figure 5.6: Density-based model representation. With the density-based model, (a) given the input data (top), a density histogram (bottom) is calculated. (b) The space is analyzed at different density values, and components are extracted. (c) Tracking components across density values leads to the creation of the merge tree, which marks topological events with nodes.

The components of the density histogram are identified by sweeping $F$, such that $0 = F_0 < F_1 < F_2 < \cdots < F_m = \infty$. At each $F_i$, histogram cells where $f_{xy} \leq F_i$ are extracted and components found by joining neighboring cells (I use the 8 surrounding neighbors). This is computed by treating histogram cells as graph nodes, $V$, iff $f_{xy} \leq F_i$. Graph edges, $E$, connect vertices that are neighbors in the density histogram, and connected components are extracted using the set union data structure with performance $O(|E|\alpha(|V|))$. Since only immediate neighbors are considered for connecting, there are $O(|V|)$ edges.
To construct the merge tree, sweeping $F_i$ from $0 \rightarrow \infty$, nodes are born at the first $F_i$, where a new component appears. As $F_i$ is increased, the components expand until they merge with another component. When components merge, the component with the more recent birth (i.e., higher $f_{xy}$) dies, while the component with the lower $f_{xy}$ continues. For example, in Figure 5.6(c), at $F_1$, the pink and purple components are about to merge. When they do at $F_2$, the pink component dies since it was born more recently (i.e., $f_{\text{pink}} > f_{\text{purple}}$), and the merged component in purple continues. Once all clusters have merged into a single component, that component dies at $\infty$ (i.e., it always exists, no matter how large $F_i$ gets).

The value of this model over the distance-based model is that it only requires the input scatterplot. It needs no information about the cluster centers, and it makes no assumptions about the distribution of points within those clusters.

5.4.3 Persistence Threshold Plot

Thus far, the models only encode the clustering structure as a function of distance or as a function of density in the merge tree. The method to select the number of clusters that will be perceived by a user is calculated similarly, irrespective of the underlying model, though the input parameters (distance vs. density) have different meanings.

For this, I generate a persistence threshold plot. For a given merge tree, each component has its persistence, $\rho$, calculated. The persistence is the difference between birth and death values of the component (i.e., $\rho = \text{death} - \text{birth}$). 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 density-based models.
Figure 5.7: The persistence threshold plots. The persistence threshold plots for the (a) distance- and (b) density-based models in Figure 5.5 and Figure 5.6, respectively. The horizontal axis represents the threshold, while the vertical axis shows the number of clusters. The red line shows how a threshold can be extracted from a given number of clusters and vice versa.

absolute scale of the feature (e.g., the absolute density value). I use persistence as a threshold to model the number of clusters a user would count in a scatterplot and vice versa.

This information is represented in a persistence threshold plot or threshold plot. To form the plot, for the threshold $T \in [0, \infty)$, at a given $T_i$, I count the number of clusters whose $\rho > T_i$. This information is encoded into the line chart (Figure 5.7) by plotting the threshold $T$ horizontally and the number of clusters vertically.

Given these functions, we have the ability to determine critical thresholds (using either model) for the visual separation of clusters. For example, the red dashed lines in Figure 5.7(b) show the persistence threshold ($T_{de}$) that corresponds to perceiving 3 clusters and vice versa. With this relationship, our models can now be used to estimate the number of clusters that a user would select in a given scatterplot.
5.5 Main Experiment

I evaluate how well the merge tree models estimate the number of clusters perceived in a scatterplot by studying 3 factors \((S, N, P)\). In addition to revisiting \([H1]\) and \([H2]\), I include 3 new hypotheses:

\[H3\] Data point size \((P)\), having a direct impact on visual density, affects the accuracy in cluster count identification in scatterplots.

\[H4\] Using a persistence threshold correlated to the distribution size \((S)\) of normally distributed clusters, the distance-based model will estimate the number of clusters perceived by users.

\[H5\] Using a persistence threshold correlated to the size of data point \((P)\), the number of data points \((N)\), and by their interaction effect \((N \times P)\), the density-based model will estimate the number of clusters perceived by users.

5.5.1 Properties and Data Generation

Using the information learned in preliminary experiment, following values were modified for the main experiment (i.e., all others remained the same, see section 5.3.1):

- Data point size/area \((P)\): \(\{1\, \text{px}, 3\, \text{px}, 5\, \text{px}, 7\, \text{px}\}\) — On the low end, 1\,\text{px} point size is the minimum possible value. On the high end, 7\,\text{px} was chosen in combination with the number of points to limit the maximum visual density to \(\sim 30\%\) of a given stimulus.

- Number of data points \((N)\): \(\{500, 2500, 12500\}\) — To decide the number of data points, I considered if data points are uniformly distributed, the maximum visual density is \(MVD = \frac{N \times a}{X \times Y}\), where \([X \times Y]\) are stimuli dimensions \([550 \times 550]\). With a target of \(< 30\%\), using \(P = 7\, \text{px}\) and
$N = 12500$ the visual density, $MVD = 0.29$, i.e., 29% of pixels filled. I noted a logarithmic effect in the preliminary experiment. Therefore, logarithmic intervals (base 5) are used.

- Distribution size ($S$): \{25_{px}, 40_{px}, 55_{px}, 70_{px}, 85_{px}\} — The distribution size was chosen to be similar to the preliminary experiment, slightly adjusted to have fixed intervals of 15_{px} between values.

The data generation process is kept similar to the preliminary experiment. A key difference is that task stimuli are generated for each participant covering all combinations of factors. For each subject, $|S| \times |N| = 15$ dataset are generated and rendered into $|15| \times |P| = 60$ scatterplot stimuli. Each participant received similar variability and the same combination of factors in their stimuli.

5.5.2 Study Procedure

This study was designed similarly to the preliminary experiment (section 5.3.2) with the following variations. Each subject was shown in stimuli from their own pool of 60 stimuli in random order, and I included a post-test questionnaire, asking participants to describe their criteria for selecting the number of clusters.

I recruited participants from Amazon’s Mechanical Turk (AMT) for the IRB approved study [33, 68]. Based upon a post hoc power analysis of the preliminary experiment data, I recruited a total of 40 participants (21 male, 19 female; ages: [18 – 64], median age group: [25 – 34]) limited to the US or Canada. 45% of participants reported having corrected vision. All participants had a HIT approval rate of $\geq 95\%$, and were compensated at US Federal minimum wage.

In total, 60 trials $\times$ 40 participants $= 2400$ responses were collected. I carried out some data quality checks on responses, and the following responses were eliminated—9 responses with task
completion time of less than 1 second and 27 responses that ran out of time—leaving a total of 2364 responses for analysis.

5.5.2.1 Suitability of Studying Point Size Using AMT

Studying visual factors, mark size in particular, on a crowdsourced environment has potential biases due to lack of control of user hardware, retinal size, viewing distance, ambient lighting, etc. For example, search task performance decreases as the viewing angle increases [93]. However, this lack of control is a commonly accepted limitation in crowdsourced studies—numerous recent AMT studies have considered mark size, among other properties, that could be impacted by this lack of environmental control, e.g., Szafir’s study of perceived color differences [280], Chung et al.’s evaluation of orderability in visual channels [56], and Kim and Heer’s study of the effectiveness of multiple visual encodings [147].

5.5.3 Analysis Methodology

I ran our data and user responses through the merge tree-based models. For the distance-based model, I first take the centers of each cluster to build the model. Then, I use the user response to the number of clusters \((U)\) to extract a persistence threshold, \(T_{di}\). After generating the threshold for all stimuli, a linear regression, using linear least squares, is calculated for \(T_{di}\) on the factor distribution size, \(T_{S_{di}}(s) = c_1 \cdot s + c_2\), where the distribution size, \(s\), is input, and \(c_1\) and \(c_2\) are calculated by the regression. Figure 5.8(a) shows the resulting regression.

The density-based model is built by using the scatterplot to generate a visual density histogram, which is the input to the model. Then, the user response to the number of clusters \((U)\) is used to extract a persistence threshold, \(T_{de}^r\). For the density-based model, multiple factors
are tested \((N, P, \text{and } N \ast P)\), each requiring their own linear regression, i.e., \(T_{de}^N(n) = c_1 \cdot n + c_2\); \(T_{de}^P(p) = c_1 \cdot p + c_2\); and \(T_{de}^{N \ast P}(n, p) = c_1 \cdot n + c_2 \cdot p + c_3\).

Threshold functions \((T_{di}^S \text{ and } T_{de}^*)\) from the merge tree are used to calculate the model-predicted number of clusters. To measure the accuracy of the user response on a given scatterplot, \(\tau\), I add new differentials, \(D_{di}^S\) and \(D_{de}^*\), for the distance- and density-based models, respectively:

\[
D_{di}^S(\tau) = U_\tau - C_{di}(T_{di}^S(\tau))
\]

\[
D_{de}^*(\tau) = U_\tau - C_{de}(T_{de}^*(\tau))
\]

where \(U_\tau\) is the user response, and \(C_{di}\) and \(C_{de}\) are the number of clusters produced by the models using a given threshold. I used the value of differentials \((D, D_{di}^S, \text{and } D_{de}^*)\) as the primary measure to analyze the effects of the factors in the cluster counting. The histograms of the differentials for both models can be found in Figure 5.9.

The study followed a within-subjects design, where all 40 subjects were exposed to all the same treatment. Hence, I use repeated measures (RM) ANOVA to analyze the effects of the factors on \(D\). For some results, due to violations of sphericity, according to Mauchly’s test, reported degrees of freedom and \(p\)-values are Greenhouse-Geisser corrected (highlighted in green) [113, 182]. Along with RM ANOVA, I calculated partial eta-squared \((\eta^2)\). As per Cohen’s guidelines for measures of \(\eta^2\): 0.01 denotes small effect, 0.06 denotes medium effect, and 0.14 denotes large effect [60].

5.5.4 Results

5.5.4.1 Model Accuracy

The distance- and density-based models both successfully estimated user perception for counting clusters. Figure 5.9 shows the performance of all models in terms of differential. From our
analysis, I observed the highest estimation accuracy was achieved using the density-model, from best to worst, $D_{de}^{N*P}$: ($\mu = 0.18$, $\sigma = 1.58$); $D_{de}^P$: ($\mu = 0.50$, $\sigma = 1.67$); and $D_{de}^N$: ($\mu = -0.53$, $\sigma = 2.14$).

The distance-based model performs next best, $D_{di}^S$: ($\mu = 1.12$, $\sigma = 2.64$). Whereas, without a model performed the worst, $D$: ($\mu = -3.74$, $\sigma = 3.00$).

### 5.5.4.2 Factor Effect Analysis Without a Model

I performed 3-factor RM ANOVA testing to analyze the factors, distribution size ($S$), number of points ($N$), and point size ($P$) in terms of the effect on the differential without a model, $D$.

I observed that the distribution size ($S$) and the number of points ($N$) had a significant effect for counting clusters with respect to the differential, $D$, with $(F_S(4, 2304) = 286.11, p < 0.001, \eta^2 = 0.32)$ and $(F_N(1.98, 1576.43) = 33.98, p < 0.001, \eta^2 = 0.029)$, respectively. On the other hand, data point size ($P$) failed to reach significance, with $(F_P(3, 2304) = 0.21, p = 0.889,$
I also tested for interaction effects and only observed a significant effect between S and N, \((F_{S \times N}(8, 2304) = 8.18, p < 0.001, \eta^2 = 0.028)\).

The \(\eta^2\) analysis showed a large effect size on distribution size (S) and a small on the number of points (N) and interaction effect \(S \times N\). This is likely because smaller distributions create denser clusters with better separation, while larger distributions blend to create ambiguous boundaries. From these results, both [H1] and [H2] are reconfirmed. The lack of significance on point size (P) indicates that [H3] should be rejected. However, I will revisit this hypothesis later.

**Figure 5.9: Histograms for user response differential against frequency.** (a) no model (skew due to users' underestimation), (b) the distance-based model, and (c) density-based models. Responses that are closer to 0 imply a good fit for the data.
Table 5.1: Distance-based model Repeated-Measures ANOVA results. These results demonstrates the effect of visual factors on the *differential* $D_{\text{di}}^S$.

<table>
<thead>
<tr>
<th>Factors</th>
<th>Df</th>
<th>F-value</th>
<th>P-value</th>
<th>$\eta^2$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution size (S)</td>
<td>4</td>
<td>552.9</td>
<td>$\leq$ 0.001</td>
<td>0.4900</td>
<td>1.91</td>
</tr>
<tr>
<td>Number of points (N)</td>
<td>1.98</td>
<td>051.39</td>
<td>$\leq$ 0.001</td>
<td>0.0420</td>
<td>2.61</td>
</tr>
<tr>
<td>Size of points (P)</td>
<td>3</td>
<td>000.41</td>
<td>0.746</td>
<td>0.0005</td>
<td>2.64</td>
</tr>
<tr>
<td>S*N</td>
<td>8</td>
<td>002.72</td>
<td>0.006</td>
<td>0.0100</td>
<td>-</td>
</tr>
<tr>
<td>S*P</td>
<td>12</td>
<td>000.13</td>
<td>1.000</td>
<td>0.0006</td>
<td>-</td>
</tr>
<tr>
<td>N*P</td>
<td>6</td>
<td>000.08</td>
<td>0.998</td>
<td>0.0002</td>
<td>-</td>
</tr>
<tr>
<td>S<em>P</em>N</td>
<td>24</td>
<td>000.21</td>
<td>1.000</td>
<td>0.0020</td>
<td>-</td>
</tr>
</tbody>
</table>

-: not calculated; Greenhouse-Geisser corrected.

5.5.4.3 Distance-based Model Factor Analysis

Using persistence threshold on distribution size, $T_{\text{di}}^S$, I calculated the *differential* ($D_{\text{di}}^S$) and performed 3-factor RM ANOVA to observe the main effects of the individual factors distribution size (S), number of points (N), and point size (P), as well as interaction effects (Table 5.1).

The analysis identified a significant effect of distribution size (S) and the number of points (N) on the *differential* ($D_{\text{di}}^S$), but the point size (P) failed to reach significance. In particular, I found a large effect for distribution size (S) on $D_{\text{di}}^S$. I also observed a small-medium effect in the number of points (N) and a negligible effect on the point size (P). I did not anticipate any interaction effects, and only $S*N$ showed a small effect. In terms of accuracy, as pointed out in section 5.5.4.1, the distance-based model improved overall accuracy over using no model (Figure 5.9(b)). Investigating further, Figure 5.8(b) shows the accuracy per distribution size. Note that the accuracy was found for all distribution sizes, except at $S = 85$, which negatively impacted overall performance. I speculate that this is due to the significant blending of distributions at this extreme. Given the large effect in S and overall improvement in accuracy, I consider 5.5 confirmed.
5.5.4.4 Density-based Model

For the density-based model, I calculate 3 variations of the threshold and differential that use the factors that most directly influence visual density. Those are the number of data points \((N/D)\), the size of data points \((P/D)\), and their interaction \((N*P/D)\). For each, I performed 3-factor RM ANOVA testing on the individual factors the distribution size \((S)\), the number of points \((N)\), and the point size \((P)\), as well as interaction effects (Table 5.2).

5.5.4.5 Histogram Resolution

The density-based model uses the visual density of a given scatterplot to model cluster perception. To calculate visual density, a 2D histogram is calculated on the image with bins of uniform width and height, \([B_{px} \times B_{px}]\). The choice of bin size for the density histogram is potentially influential in our analysis, as bins that are too small may cause instability, and bins that are too large may miss clusters. To determine the appropriate bin size, I performed an analysis on the data from Figure 5.3. A set of stimuli images are generated with fixed values for factors \((C = 6, N = 2500, and P = 7_{px})\) and different values for \(S = \{25_{px}, 40_{px}, 55_{px}, 70_{px}, 85_{px}\}\). I plotted the normalized density threshold (i.e., density threshold divided by area of a bin, i.e., \(T_{de}/B^2\)) generated

<table>
<thead>
<tr>
<th>Model</th>
<th>Grid</th>
<th>Distribution Size (S)</th>
<th>Number of Points (N)</th>
<th>Point Size (P)</th>
<th>Interaction (N*P)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D^N^P_de</td>
<td>10_{px} \times 10_{px}</td>
<td>4 061.45 ≤ 0.001 0.090</td>
<td>2 122.60 ≤ 0.001 0.090</td>
<td>3 086.28 ≤ 0.001 0.099</td>
<td>6 11.28 ≤ 0.001 0.028</td>
</tr>
<tr>
<td>D^S^P_de</td>
<td>10_{px} \times 10_{px}</td>
<td>4 014.99 ≤ 0.001 0.020</td>
<td>2 058.80 ≤ 0.001 0.045</td>
<td>3 301.80 ≤ 0.001 0.280</td>
<td>6 10.20 ≤ 0.001 0.025</td>
</tr>
<tr>
<td>D^N^P_de</td>
<td>10_{px} \times 10_{px}</td>
<td>4 129.22 ≤ 0.001 0.180</td>
<td>2 216.00 ≤ 0.001 0.150</td>
<td>3 613.20 ≤ 0.001 0.430</td>
<td>6 56.90 ≤ 0.001 0.120</td>
</tr>
<tr>
<td>D^S^P_de</td>
<td>10_{px} \times 10_{px}</td>
<td>4 088.58 ≤ 0.001 0.130</td>
<td>2 096.50 ≤ 0.001 0.075</td>
<td>3 006.00 ≤ 0.001 0.086</td>
<td>6 1.62 ≤ 0.001 0.004</td>
</tr>
<tr>
<td>D^N^P_de</td>
<td>10_{px} \times 10_{px}</td>
<td>4 004.98 ≤ 0.001 0.008</td>
<td>2 143.80 ≤ 0.001 0.100</td>
<td>3 074.30 ≤ 0.001 0.080</td>
<td>6 11.40 ≤ 0.001 0.028</td>
</tr>
<tr>
<td>D^S^P_de</td>
<td>10_{px} \times 10_{px}</td>
<td>4 094.45 ≤ 0.001 0.130</td>
<td>2 439.70 ≤ 0.001 0.270</td>
<td>3 135.40 ≤ 0.001 0.140</td>
<td>6 010.80 ≤ 0.001 0.250</td>
</tr>
<tr>
<td>D^{N*P}_de</td>
<td>10_{px} \times 10_{px}</td>
<td>4 057.95 ≤ 0.001 0.090</td>
<td>2 383.90 ≤ 0.001 0.240</td>
<td>3 248.30 ≤ 0.001 0.240</td>
<td>6 130.30 ≤ 0.001 0.250</td>
</tr>
<tr>
<td>D^{N*P}_de</td>
<td>10_{px} \times 10_{px}</td>
<td>4 007.56 ≤ 0.001 0.012</td>
<td>2 047.30 ≤ 0.001 0.038</td>
<td>3 061.20 ≤ 0.001 0.070</td>
<td>6 11.40 ≤ 0.001 0.028</td>
</tr>
<tr>
<td>D^{N*P}_de</td>
<td>10_{px} \times 10_{px}</td>
<td>4 062.81 ≤ 0.001 0.096</td>
<td>2 273.30 ≤ 0.001 0.180</td>
<td>3 195.50 ≤ 0.001 0.190</td>
<td>6 19.10 ≤ 0.001 0.046</td>
</tr>
</tbody>
</table>

The \([20_{px} \times 20_{px}]\) grid is highlighted to indicate it is the primary focus of our analysis.
by $U = 6$ clusters for 5 different bin sizes (Figure 5.10). The results showed instability in the density threshold for smaller values and a stable result starting at $[20_{px} \times 20_{px}]$. For this reason, 3 resolutions of histogram cell sizes are reported: $[10_{px} \times 10_{px}]$, $[20_{px} \times 20_{px}]$, and $[40_{px} \times 40_{px}]$, but our main discussion focuses on $[20_{px} \times 20_{px}]$.

### 5.5.4.6 Number of Points Model ($T^{N}_{de}/D^{N}_{de}$)

RM ANOVA results demonstrate significant and consistent main effects of $S$, $N$, $P$, and interaction effect of $N \ast P$, which can be seen in Table 5.2. Point size has a large effect on $D_{de}$, confirming our hypotheses and previous work (e.g., [242]) of density’s influence on cluster perception. The number of points showed a small-medium effect size on $D_{de}$, also align with our hypotheses. The accuracy of the number of points model was the worst of the 3 density models, though still significantly better than no model (Figure 5.9(c)). The accuracy of the model, plotted by the number of points in Figure 5.11(a), shows lower accuracy as the number of points increases.

![Figure 5.10](image.png)

Figure 5.10: Normalized persistence threshold for the density-based model for the data. The plot is based on Figure 5.3 which shows stability at histogram bins size $[20_{px} \times 20_{px}]$ and larger.
5.5.4.7 Point Size Model \( (T_{de}^P / D_{de}^P) \)

In this model, the number of points showed a medium-large effect size, while point size demonstrated a medium effect size for the differential. On the other hand, interaction of \( N \times P \) results small values of \( \eta^2 \) (Table 5.2). The overall accuracy of this model was better than the number of points model (Figure 5.9(c)). Figure 5.11(b) shows the accuracy per point size. The model was largely accurate, except for the smallest size, \( P = 1_{px} \).

5.5.4.8 Interaction Model \( (T_{de}^{N \times P} / D_{de}^{N \times P}) \)

Similar to the previous two models, significant effects were observed for all factors. However, only point size demonstrated medium effect size (Table 5.2). This model showed the best overall accuracy of any model tested (Figure 5.9(c)). This makes logical sense, as the density is the combination of the number of points and their size. Figure 5.11(c) shows the accuracy per number of points and per point size. For both cases, the accuracy was improved. However, \( P = 1_{px} \) was still the worst performing category.

Our analysis showed the number of points, point size, and their interaction all had significant effects and improved accuracy over no model. Therefore, I consider \( [H5] \) confirmed. Furthermore, I identified some large effects with point size for the density-based model, and this indirect relationship confirms\( [H5] \).

5.5.4.9 Post-Test Questionnaire

To further support our hypotheses, I asked the participants to state the criteria that influenced their counting of clusters in a free-response format at the end of the experiment. Their responses
Figure 5.11: Density-based model mean and 95% CI on factors. The confidence interval plots of user response differential, $D_{de}^*$, on factors with density histogram [20 px × 20 px].

largely mirrored our findings—10% cited the size of symbol; 25% responses cited something amounting to distribution size; 25% cited distance between clusters; and 65% of responses included density as a factor.\(^{15}\)

5.6 Model Usage

Identifying clusters is an important low-level visual analytics task [10], as well as in data analysis in general [202]. Still, as mentioned in section 5.1, clustering is an ill-posed problem, with the “correct result” being subject to the constraints of the algorithm or individual performing the analysis. Through our evaluation, I have shown that our models, the density-based model, in particular, performed well in estimating the number of clusters an average human would perceive. However, this in and of itself is not the real application value of the models. Instead, the models can be used to optimize the visual encodings to maximize the saliency of the visualization. Furthermore, the threshold plots provide an evidence-based rationale for design decisions.

\(^{15}\)Some subjects listed multiple criteria.
Table 5.3: Summary of the main effects found in the study based on $\eta^2$. The colors are the same as in Figure 5.9.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Without Model</th>
<th>Distance-based ($T_{di}$)</th>
<th>Density-based ($T_{de}^N$)</th>
<th>($T_{de}^P$)</th>
<th>($T_{de}^{N*P}$)</th>
<th>($T_{de}^O$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution size (S)</td>
<td>□□□</td>
<td>□□□</td>
<td>□□□</td>
<td>□□□</td>
<td>□□□</td>
<td>□□□</td>
</tr>
<tr>
<td>Number of points (N)</td>
<td>□□□□□□</td>
<td>□□□</td>
<td>□□□</td>
<td>□□□ □□□</td>
<td>□□□ □□□</td>
<td>□□□</td>
</tr>
<tr>
<td>Size of points (P)</td>
<td>□□□□□□</td>
<td>□□□</td>
<td>□□□</td>
<td>□□□ □□□</td>
<td>□□□ □□□</td>
<td>□□□</td>
</tr>
<tr>
<td>Opacity (O)</td>
<td>□□□□□□</td>
<td>□□□</td>
<td>□□□</td>
<td>□□□ □□□</td>
<td>□□□ □□□</td>
<td>□□□</td>
</tr>
<tr>
<td>Interaction</td>
<td>□□□□□□ □□□□□□</td>
<td>□□□□□□ □□□□□□</td>
<td>□□□□□□ □□□□□□</td>
<td>□□□□□□ □□□□□□</td>
<td>□□□□□□ □□□□□□</td>
<td>□□□□□□ □□□□□□</td>
</tr>
</tbody>
</table>

large effect; medium effect; small effect; negligible effect; not tested

5.6.1 Controlling Design Factors

The models I have introduced construct a bridge for visualization designers between their choice of visual encodings and how users perceive clusters. Table 5.3 summarizes our findings and suggests how designers should focus their design decisions on selecting visual properties that robustly support cluster identification in scatterplots. For the distance-based model, *distribution size (S)* was the only factor with a large effect size. On the other hand, with the density-based model, the *number of points (N)*, *point size (P)*, and *opacity (O)* all showed large effects on cluster count perception.

1. Distribution Size — Visualization designers generally do not have control over distribution size (S) in the data. Although distributions are rarely known a priori, they can be extracted from scatterplots, e.g., using Gaussian mixture models, which, combined with the distance-based model, could be used to help the designer to understand the number of clusters a user is likely to see. Nevertheless, this approach is unlikely to be helpful to the majority of designers.
2. **Number of Points** — Visualization designers have *limited control* of the number of points (N), mostly in terms of data subsampling, e.g., [50, 128, 156], which influences visual density. Using uniform random sampling, e.g., [76, 89], or targeted nonuniform sampling, e.g., by using density [29], can reduce the number of points and, thus, the visual density. The density-based model can be used to evaluate what level of sampling provides optimal saliency of clusters.

3. **Point Size** — The point size (P) is the first design factor with *complete control* in scatterplots. As pointed out earlier, increasing the area of pixels also increases the visual density. Once again, the density-based model can be used to help select the point size that provides the optimal saliency. There is an important interplay between the number of points and the point size, as adjusting either can influence the visual density.

4. **Opacity** — Opacity (O) is another factor for which the designer has *complete control*, from fully transparent to fully opaque, once again impacting the visual density of the scatterplot. As suggested by Urribarri and Castro, when selecting opacity, there is a trade-off with picking a point size [290] and, given our analysis, also with the number of data points shown. Nevertheless, using the density-based model, various opacity levels, along with the number of points and their size, can be evaluated and the optimal configuration selected.

5.6.2 Threshold Plots for Optimizing Cluster Saliency

As our goal is to improve the effectiveness of the visualization design, 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. Using pre-studies of the effects of different visual encoding configurations in scatterplots, visualization
practitioners can pick the configuration that maximizes the visibility of clusters. The web version of cluster perception framework is available as TopoCluster: <https://usfdatavisualization.github.io/TopoClusterPerceptionDemo>.

Consider Figure 5.12, for example. The 3 scatterplots are each plotted in the persistence threshold plot. The horizontal axis reveals for each plot how salient each of the clusters are. The 10% opacity plot shows between 1 and 8 clusters are visible, but either 2 or 6 clusters are most visible. That is not to say other numbers of clusters are not visible, but they are simply not as distinctive. Of the 3 scatterplots in Figure 5.12, 10% provides the best saliency, followed by 1%, then 100% opacity.

5.6.3 Case Study

To demonstrate the utility of the models on real data, I showed how the choice of visual encoding impacts the cluster perception. I performed a case study using dimensionality reduction on the MNIST dataset [159], which is an extensive database of handwritten letters commonly used to test machine learning techniques. Here I explore the dataset, which consists of 70K samples with 10 labels of handwritten digits (the labels are not used in sampling or rendering). I applied both t-SNE

![Figure 5.12: Example of overplotted scatterplot stimuli. The factors are C = 11, N = 200,000, S = 55, P = 7 px, but varying opacity, 1% (blue), 10% (green), and 100% (orange).](image)
(Figure 5.13(a)) and PCA (Figure 5.13(b)) to plot the features on a 2D scatterplot and demonstrate the influence of two factors, the number of points and opacity.

In Figure 5.13(a), I show the results of varying the number of points (after dimension reduction), where $N = \{500, 2500, 12500\}$, by using random sampling. The resulting persistence threshold plot shows that for $N = 500$, in red, clusters are difficult to differentiate. For $N = 2500$, in blue, and $N = 12500$, in purple, both have similar levels of effectiveness, with 12500 having a slight advantage, making it the better choice for representing this example.

In Figure 5.13(b), I show the results of varying the opacity of the data points, where $O = \{1\%, 5\%, 10\%, 50\%, 100\%\}$. The results in the persistence threshold plot fall into 3 groups. On
the first extreme, $O = 50\%$, in purple, and $O = 100\%$, in orange, provide no differentiation of any clusters. On the other extreme, $O = 1\%$, in blue, shows that a relatively low level of saliency for 1, 2, or 3 clusters. The final group, $O = 5\%$, in pink, and $O = 10\%$, in green, both show identically high levels of saliency for 1, 2, or 3 clusters in the data, making either of these the better choice for representing this example.

5.7 Discussion and Conclusions

Scatterplots are a common type of visualization, used to identify clusters in datasets. In this work, I tested and validated the importance of 4 visual factors—distribution size, the number of data points, the size of data points, and the opacity of data points—in cluster perception and built 2 models: a distance- and density-based model for the task. Our results confirm the theoretical models of Sadahiro, which states data points distribution (proximity), and number and size of data points (concentration and density change) affect cluster perception [242]. Finally, our findings confirm the important role that the choice of visual factors can have on cluster identification—visualization practitioners may apply these models for optimizing properties of their visualizations.

5.7.1 Model Limitations

Both of our models have some limitations. In the distance-based model, I required knowledge of the centers of the clusters with a fixed-size isotropic normal distribution for the model—considering other distributions would likely require modifications to the model. This requirement is particularly restrictive with respect to non-synthetic data. I showed that user response accuracy in the density-based model was significantly better than the distance-based model. However, choosing the correct density histogram resolution is a critical task that may also be dependant on the data. A choice of
an extremely high or low resolution could reduce the accuracy of the threshold value. Additionally, although the density model does not directly consider a normal distribution, I have only tested it against fixed-size normal distributions. Using the model with other types of distribution should be treated with caution.

5.7.2 Alternative Models

Alternative models could potentially be developed to similarly explain the variance. With respect to distance, hierarchical clustering could be used, which is functionally equivalent to our distance model. For density, since stimuli are built on Gaussian distributions, a Gaussian Mixture Model (GMM) could be used. GMMs, being numerically extracted, cannot provide the same theoretical guarantees as our models, which are technically combinatorial. The theoretical guarantees, coming from persistent homology, also include *stability* guarantees. With stability, small changes in the input are guaranteed to produce only small changes to the output. A consequence of stability is robustness to noise. The noise has low persistence, not influencing the selection of the number of clusters.

5.7.3 Automatic Parameter Optimization

One natural extension of this work is to develop a (semi-)automatic model for selecting design factors for a dataset. Unfortunately, using threshold plots as-is represents an under-constrained optimization, and it requires, *at the very least*, a user specification of the number of clusters in the data.
Chapter 6: Scatterplot Design Optimization Using Clustering

6.1 Introduction

Scatterplot are intuitive and widely used visualization [104] to reveal relationships and patterns between quantitative variables, e.g., correlations [121], clusters [188], exploring outliers [248], etc. Scatterplots encode bivariate data using different encoding methods [58], and studies have demonstrated the effect of visual encoding as well as data aspects on visual tasks [147, 220]. Several perceptual studies have evaluated the effectiveness of scatterplots in low-level tasks that include measuring a viewer’s ability to assess trends [196], correlation perception [230], and average values and relative mean judgments [109], and clustering [220]. Studies have focused on how scatterplot designs can be optimized in order to render an effective visualization, e.g., visual encodings including data point size [147], number of data points [109], opacity [186], color [280], shape [256].

Design optimization using a visual encoding, such as data point size or opacity, and data aspect, such as the number of data points or sampling rate, improve user performance on low-level visual tasks [10], data interpretation, and communication. Design choices in scatterplots, such as the graphical encodings, or data aspects, can directly impact the quality of decision making for the low-level tasks such as clustering. Clustering occurs when patterns in the data form distinct groups [10, 248]. Effective visualization design improve understanding of data by leveraging visual perception to enhance understanding. Hence, constructing frameworks that consider both the perceptions of the visual encodings and the task being performed enables optimizing visualization
design to maximize efficacy. However, we are still missing a framework that provides an optimized design framework for data clarity or task efficacy on scatterplots, effective cluster perception in our case, using visual encodings, and data aspects.

Overplotting in scatterplot directly influences the visual density of the display and obscures the underlying data patterns. Visual designers control visual factors such as point size and opacity, whereas data distribution is inherent, and designers have no control over it. On the other hand, designers have limited control over the number of data points, primarily via data subsampling. Data subsampling is usually applied to reduce the clustering in overplotted scatterplot along with reducing mark size or opacity values [92]. An optimal design optimization must consider the associated determinants of overdrawing solution in the scatterplot, which is the number of data points or sampling rate, recommending the designer various design choices options for a scatterplot design.

I present work demonstrating design optimization in scatterplot visualization filling the missing gap between guidelines and application. I import perceptual design guidelines on visual encoding and data aspects in a scatterplot and introduce a user-guided optimization model to present an optimal scatterplot design that suggests the parameters for the factors influencing the visual density and clutter reduction technique; data point size, opacity, number of data points, and sampling algorithm. The proposed optimal design solution focuses on the clustering task in a scatterplot and discusses how the designer, and practitioner can extend it to other low-level tasks.

The interactive tool in this chapter leverages the application of the threshold plot introduced in the previous chapter (chapter 5). The threshold plot is calculated on the visual density to optimize the design decisions on data point size, opacity, sampling rate (number of points), and sampling algorithms. The pipeline of model works as follows: data is subsampled; scatterplot is rendered;
threshold plot is generated for blurred scatterplot images using visual density; and finally, the cluster structure for given scatterplots are ranked to provide the optimal solution. The highest rank design from the model is considered the optimal design choice for showing salient cluster structure. The model is validated using a user study that found a threshold plot as a proxy for cluster structure saliency captures an optimal scatterplot design. The user study was conducted with 70 participants from AMT, and the results confirmed and validated the model optimized design choices. The effect was particularly pronounced when the value for the scatterplot threshold bar length was high. Further, a case study suggests that our model requires less interaction and time to select an optimal design. Our proposed tool will work in two ways: 1) provide the scatterplots with salient cluster structure for the given number of clusters based on the visual factors, and 2) the interactive model results can be used as an optimized design choice.

The primary contributions of this chapter are an optimization model and user-guided dynamic parameter selection tool for optimizing scatterplot design based upon the data. The loci of this study are clustering on a scatterplot, with guidelines and available prototypes that can be extended to other low-level tasks. The model results are validated on user study against user understanding, and findings suggest our model can be used as a proxy for cluster structure saliency. Finally, the open-source web interface tool can be used to dynamically rank the cluster structure's saliency in scatterplot using data aspects (subsampling algorithm and sampling rate) and visual encodings.

6.2 Prior Work

I provide brief coverage of clustering in scatterplot, visual factors in scatterplot design, solution to clutter reduction, and design optimization on scatterplot.
6.2.1 Clustering in Scatterplots

Clustering and segmentation is “grouping of similar data points on scatterplot in a given dataset” [10] to reveal characteristics of data and allow visualization designers and practitioners to further explore underlying patterns and trend [247, 248]. Another definition of clustering as suggested by design factor survey study, is a high-level data characterization with the ability to identify groups of similar items [248]. Data features, such as segmentation and clustering, have been studied in different types of visualization in data analysis contexts, such as text [7], maps [165, 304], and bubble charts [281]. Previous works investigated modeling the perception of clusters in scatterplots. Matute et al. technique quantified and represented scatterplots through skeleton-based descriptors measuring scatterplot similarity [181]. However, their approach does not consider visual encodings in the evaluation. ScatterNet, a deep learning model, captures perceptual similarities between scatterplots that could be used to emulate human clustering decisions [176]. The scagnostics technique focused on identifying patterns in scatterplots, including clusters [69]. However, later a study by Pandey et al. proved they do not reliably reproduce human judgments [206].

6.2.2 Design Optimization in Scatterplots

Rensink’s framework for reasoning about perceptions of visualization designs suggests using techniques from vision science [229]. The extended-vision theory asserts that a viewer and visualization system is a single system, whereas the optimal-reduction thesis postulates an optimal visualization. The work focuses on the fundamental questions could we determine if its design is optimal?
Recently, ClustMe used visual quality measures to model human judgments to rank scatterplots [1]. ClustMe performed well in reproducing human decisions for cluster patterns. Further, threshold plot generated using visual density are used to predict the number of the cluster using given values of design factors [220]. In this chapter, I extend the application of threshold plots to choose the optimal design choices for a salient cluster structure in a scatterplot. I aim to take one step next to the design guideline stage, where a prototype utilizes the findings. This work can be extended to other low-level tasks in a scatterplot. Optimization studies focused on other aspects of a 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 [300]; automatically selecting the optimal representation between scatterplot and line graph for trend exploration in time series data [301]; 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 [186].

6.2.3 Visual Factors in Scatterplots Designs

Data are encoded on the screen using data symbol and size of those data points in a scatterplot and acts as an important factor in visual aggregation tasks in scatterplots [281]. As the size of data points on the scatterplot increases, so does the density, which directly influences cluster perception [242]. Symbol size also has a direct influence on discriminability in certain tasks [163], e.g., in color perception tasks [275]. Szafir’s study on color-difference perception found that perceived color difference varies by the size of marks [280]. Size also influences search task effectiveness as increasing symbol size reduces search time in a display of random points [111]. Another critical factor in data encoding is opacity. Reducing mark opacity can alleviate overplotting to assist visual
analytics tasks [220, 247], e.g., spike detection in dot plots [67]. In this model, point size and opacity are one of the visual encodings to optimize the design choice.

The quantity of data points on the screen directly influences the visual density and overdrawing of a scatterplot. Also, the higher the number of points in a given area, the higher the chances are that they would be perceived as a cluster due to increased density [242]. Gleicher et al.’s empirical study asked participants to compare and identify average values in multi-class scatterplots [109]. 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 [220]. Reducing the number of points reduces the overplotting and reveals underlying pattern [92].

On the other hand, visual encoding properties, such as mark type, mark size, and mark opacity, have the potential to influence perceptual judgments [58]. Additionally, a few of these mentioned data aspects and visual encoding can be manipulated to solve overdrawing to some extent. In a recent study [220], data aspects, including data distribution size/type and the number of data points, and visual encoding properties: mark size, and mark opacity are investigated to understand how these various factors influence the perception of clusters in scatterplot. This is particularly notable since many of these factors are under the control of the visualization designer, and optimized design choices would lead to correct, less distorted, or unbiased presentation and provide an accurate summary and conclusions of data. In this work, I am applying the perceptual study findings on visual encodings and data aspects on a design optimization tool where I select optimal design choice values on a scatterplot.
6.2.4 Overdrawing in Scatterplots and Solutions

Scatterplot’s performance to reveal underlying data pattern suffer from data points overplotting making it inefficient for data analysis. Overplotting in scatterplot directly influences the visual density of the visual display and obscures the underlying data patterns. A taxonomy of clutter-reduction techniques [92] listed various category and examples focusing on data display; such as varying point size [20, 72, 214, 322], varying opacity level [99, 136, 152, 180, 309], pixel-plotting [135, 143, 225], and reducing the number of data points through subsampling [48, 60, 128, 290].

The three of the useful strategies are (1) reducing the number of data points through subsampling; (2) reducing the opacity; and (3) reducing the point size. This study will be mainly focused on two visual encodings point size and opacity values.

- Reducing Point Opacity: Reducing mark opacity can alleviate overplotting to aid various visual analytics tasks [247], e.g., spike detection in dot plots [67]. The optimal value of alpha provides the right amount of transparency to discern the difference in the amount of overplotting. Furthermore, varying opacity levels aid in different visual tasks—while low opacity benefits density estimation for large data, it also makes locating outliers more difficult [186]. Matejka et al. defined an opacity scaling model for scatterplots based on the data distribution and crowdsourced responses to opacity scaling tasks [180]. 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 [92]. Further, making the points more transparent would be less helpful when there many points.

- Reducing Point Size: Scatterplot designs where larger points are used may obscure the visibility of underneath points, and hence reducing the point size would be beneficial. But reducing
the point size would conflict with color-based encoding on the data points as color difference varies with point size [280]. Furthermore, in the case of relatively minor overplotting, reducing point size would be helpful, but when point size is already too small, this method cannot be implemented [103].

- Applying New Technique: Keim et al. developed a method called the generalized scatterplot, which allows users to strike a balance between overplotting and distortion [144]. Other developed methods employed the fundamentals of density estimations [253]. Carr et al. used hexagonal cells to accumulate densities [43]. Bachthaler and Weiskopf created a continuous density field by using a rigorous, accurate, and generic mathematical model to produce the continuous scatterplots [16, 17]. A recent study worked on enhancing the cluttered scatterplots, called Sunspot Plots, demonstrating a smooth blending of discrete and continuous representations enables the visualization of leading trends in dense areas while still preserving outliers in sparse regions [286]. Mayorga and Gleicher proposed Splatterplots to highlight outliers as discrete markers and dense regions as smooth contours with the application of color mapping and contouring explicitly [183]. Unfortunately, color blending could degenerate and synthesize new colors that are difficult to interpret, shown as a limitation to splatterplots [183].

- Data Subsampling: The classical way of reducing the number of points in random sampling uses the approach of randomly selected data points. Ellis and Dix used random sampling to visually reduce data density [76, 89]. Selecting points by simple random sampling will void out data patterns in low-density areas showing misleading representations. 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 [28]. An improvement of the random method is a non-uniform sampling that treats different parts of the scatterplot in different ways with an aim to preserve the relative density [27], which is further extended to understand the perceiving density difference [29]. Data sampling selected a subset of data points from the original data sets and displayed them to reduce the overdraw [92]. A comprehensive evaluation study on different sampling methods comprised of four experiments investigated the effects of sampling methods on 2D scatterplots [334]. Their study mainly focuses on the properties of the techniques such as density preservation, outlier detection, and spatial separation. In this work, in terms of data aspects I am studying number of points (derived from sampling rate) and sampling algorithm. With recent work on an overdrawn solution in the scatterplot, I have observed the results as more guidelines and what should be done to subdue the clutter as a challenge to a particular issue (visual density or identifying pattern).

6.3 Methods

Visualization effectiveness, a task-dependent engagement directly impacting the design choices, is a constant pursuit to ameliorate efficacy. An optimized design selects the optimal choice from the possible visualization design to effectively convey the variation in the data and the design. An optimized design in data visualization is critical to an unbiased and valid conclusion. Our primary objective of the study is to provide an optimized design choice for a scatterplot that will make it effective for cluster structure saliency. This particular model will also incorporate the solution of overdrawing using data subsampling in the scatterplot, blurring, and other primary visual encodings—point size and opacity. Under this section, I will discuss methods of data sampling, visual encoding, blurring, threshold plot and their application in our proposed model.
6.3.1 Overview

The proposed optimization model in this chapter allows us to interactively choose the optimal design choice using a user-guided automatic parameterization which uses threshold plot introduced in chapter 5. The idea behind the model is — how our model acts as a proxy to measure/estimate the cluster structure saliency based on the scatterplot design and factors. The optimized parameterization of a scatterplot consists of data aspects: number of data points, and visual encodings: data point size and opacity. This work focuses on how we can have a user-guided optimization for the parameters required to create a visualization that improves the data clarity and leads to a better decision-making task, clustering on scatterplot in our case. The output from the interactive model is a set of scatterplots in the descending order of ranking, representing the cluster saliency value.

To provide an abstract view of our entire optimization working model, I outlined the stages under the Figure 6.1 as (1) dataset, (2) projection, (3) feature extraction, (4) optimized design.

1. Dataset: Eight overdrawn scatterplot examples are the input to the model.

2. Projection: This stage comprises of following sub-stages.

   • Sampling: Dataset subsampled to reduce the overdrawning using different algorithm.
   • Encoding: Data point size and opacity values are used to encode the points.
   • Rendering: Scatterplots are rendered.
   • Blurring: Scatterplot image blurred to obscure noise.

3. Feature Extraction: This stage calculates the visual density for the scatterplot. Visual density is used in the model to compute and construct threshold plots.
4. Optimized Design: Finally, the threshold bars are used to identify the salient cluster structure as the optimized design choice.

6.3.2 Data Sampling

Subsampling aids in decreasing the visual clutter by reducing the number of data points while keeping the original structure of the data in mind. The data subsampling inputs the original dataset and selects a subset of actual data points representing the entire dataset. The subset data samples must reflect the properties of original representation to offer a fair and correct representation of the originals dataset. I organize the subsampling techniques with a taxonomy based on the properties an algorithm is preserving. Reviewing the prior work, I observed properties each algorithm preserves in the subset data. These preserved properties represent a similar distribution to the actual dataset: spatial separation, relative visual density, outliers, and random.

6.3.2.1 Random Sampling

Random sampling is a classical single-class sampling method widely used by data scientist to reveal underlying insights [234]. Random sampling does not require any prior knowledge of the data.
set, and that can be disadvantageous to sampled subset since local or global trends may disappear, sampling artifacts could be introduced, or important outliers may not be preserved. Also, multiple sampling strategies on the same dataset would enable more effective evaluation of datasets [140]. Clutter reduction studies by Ellis and Dix [76, 89] employed random sampling.

Random sampling employs a uniform sampling strategy treating all samples equally and selects each sample with the same probability, and it is easy to implement. It also preserves the intensity of density differences, but to reveal density differences in very crowded zones requires a sample ratio that destroys data in faint areas.

Table 6.1: Sampling algorithm implemented in this study and their application. Category color-coding denotes the feature preservation.

<table>
<thead>
<tr>
<th>Sampling Methods</th>
<th>Application</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Sampling</td>
<td>[79, 216, 232, 325]</td>
<td></td>
</tr>
<tr>
<td>Density Biased Sampling</td>
<td>[204, 326]</td>
<td></td>
</tr>
<tr>
<td>Non-uniform Sampling</td>
<td>[27, 29]</td>
<td></td>
</tr>
<tr>
<td>SVD Based Sampling</td>
<td>[138]</td>
<td></td>
</tr>
<tr>
<td>Multi-view Z-order Sampling</td>
<td>[128]</td>
<td></td>
</tr>
<tr>
<td>Recursive subdivision Sampling</td>
<td>[50]</td>
<td></td>
</tr>
<tr>
<td>Outlier Biased Density Sampling</td>
<td>[326]</td>
<td></td>
</tr>
<tr>
<td>Outlier Biased Random Sampling</td>
<td>[168, 337]</td>
<td></td>
</tr>
<tr>
<td>Hashmap Based Sampling</td>
<td>[51]</td>
<td></td>
</tr>
<tr>
<td>Outlier Biased Blue Noise Sampling</td>
<td>[326]</td>
<td></td>
</tr>
<tr>
<td>Multi-class Blue Noise Sampling</td>
<td>[48, 311]</td>
<td></td>
</tr>
<tr>
<td>Blue Noise Sampling</td>
<td>[48, 310, 326]</td>
<td></td>
</tr>
<tr>
<td>Farthest Point Sampling</td>
<td>[24]</td>
<td></td>
</tr>
<tr>
<td>Z-order Sampling</td>
<td>[128, 338]</td>
<td></td>
</tr>
</tbody>
</table>

- Preserving spatial separation
- Preserving relative visual density
- Preserving outliers
- Random
6.3.2.2 Preserving Spatial Separation

There are some cases where subsets of data points are required to be better spatially separated to represent the underlying data patterns and preserves spatial separation between class/clusters or highly dense regions.

Methods under this category are as follows: Non-uniform sampling strategies assign varying sampling probability to data so that some specific properties of the original datasets can be better preserved. Blue noise sampling achieves this by selecting samples with blue noise properties so that the selected samples will distribute evenly in the sample space [310, 329]. Blue noise sampling inspired by [332] contains the samples that are randomly located but remain spatially uniform. Farthest point sampling select samples with better spatial separation. It randomly picks the first sample and then iteratively selects samples of maximal minimum distances to the previously selected ones [24].

Blue noise sampling [310] can be extended to multi-class scenarios to create multi-class blue noise sampling to maintain the blue noise properties of each class of samples and of the whole dataset [311]. Based on the multi-class blue noise sampling, Chen et al. employed a hierarchical sampling strategy that selects samples round by round [48]. It first selects samples from the coarsest level using multi-class blue noise sampling. When the selected samples are not enough, it reduces the restricted distance of the chosen samples by half and adds more samples in the final result.

The notable feature of this category is maintaining the spatial distribution and separation, which helps the analysts identify the underlying patterns such as clustering or distribution of data points after the subsampling process. However, the algorithms are time-intensive, and sampling time increases with the number of data points, e.g., blue noise sampling.
6.3.2.3 Preserving Relative Visual Density

The sampling method under this category aims at preserving the relativity in the visual density of both dense and sparser regions. These approaches aim to reduce the density in highly occluded regions while preventing a further decrease in the density of sparse areas, which would result in information loss. In this category, the algorithm sample the data preserving, density-related features.

We define *density preserving* [204] as mean of the expected sum of the weights of each group’s sampled points is proportional to the group’s size. We have \( n \) values \( x_1, x_2, ..., x_n \) partitioned into \( g \) groups that have sizes \( x_1, x_2, ..., x_g \). If group \( i \) contains the points \( x_1, x_2, ..., x_n \), point \( x_j \) is included in the sample (with weight \( w_j \)) with probability \( P(x_j) \), then density preservation for \( n \) values can be defined as:

\[
\sum_{i=1}^{n_i} x_i P(x_j) = Kn_i
\]

for some constant \( \kappa \). This formalizes the notion of “representative of the data distribution”. Uniform sampling satisfies this definition. A p-uniform sample has \( \kappa = p \).

- Visual Density: *Visual (data) density* refers to the ratio between the number of displayed data samples and the corresponding area in the data rendering area. In a given region \( \Omega \), *visual density* of \( \Omega \) can be defined as the proportion of nonempty space inside \( \Omega \):

\[
D^0_V(\Omega) = \frac{\sum_{X \in \Omega^d(D^0_V(X) \neq 0)}}{|\Omega|}
\]

(6.2)
\[ D^i_v(\Omega) = \frac{\sum_{X \in \Omega} \delta(D^i_v(X) = 0)}{|\Omega|} \] (6.3)

where \(|\Omega| = \sum_{X \in \Omega} 1\) denotes the area of \(\Omega\) and \(\delta\) is an indicator function which returns one, if the condition is true, and zero otherwise.

- Relative Data Density: Bertini and Santucci proposed the concept of *relative data densities*, which is defined on the region level [29]. Given two regions \(\Omega A\) and \(\Omega B\) with the same area, the relative data density between them is defined in Equation 6.5.

\[
\phi(D^0(\Omega_A), D^0(\Omega_B)) = \begin{cases} 
1 & \text{if } (D^0(\Omega_A) > D^0(\Omega_B)), \\
0 & \text{if } (D^0(\Omega_A) = D^0(\Omega_B)), \\
-1 & \text{if } (D^0(\Omega_A) < D^0(\Omega_B))
\end{cases}
\] (6.4)

- Relative Class Density: The concept of relative densities between classes has limited exploration. Chen et al. qualitatively evaluate multi-class sampling results using relative densities [48]. Based on the definition from [50], and in context with relative data densities, *relative class densities* between two distinct classes; \(i\) and \(j\) \((i \neq j)\) on common area is defined in Equation 6.4 where the comparison result is also weighted by the class density.

\[
\phi(D^i(\Omega), D^j(\Omega)) = \begin{cases} 
1 & \text{if } (D^i(\Omega) > D^j(\Omega)), \\
0 & \text{if } (D^i(\Omega) = D^j(\Omega)), \\
-1 & \text{if } (D^i(\Omega) < D^j(\Omega))
\end{cases}
\] (6.5)
Preserving relative data and class densities has to be done in local regions. If $\phi(D_0^\Omega(\Omega_A), D_0^\Omega(\Omega_B))$ equals $\phi(D_0^\Omega(\Omega_A), D_0^\Omega(\Omega_B))$, then in that case output visualization preserves the relative data density between regions $\Omega_A$ and $\Omega_B$. In the similar manner, if $\phi(D^i(\Omega), D^j(\Omega))$ equals $\phi(D^i(\Omega), D^j(\Omega))$ then in that case output visualization preserves the relative class density between $i$th and $j$th class in region $\Omega$.

Methods under this category are as follows: *Density biased sampling* features the visual density in the data as the important factors and tends to probabilistically over-sample sparse regions, and under-sample dense regions in the sample space [204]. It can counterbalance samples from both regions, thus preserving small clusters and more solitary samples.

Density Biased Sampling is related to sampling techniques called Probability Proportional to Size (PPS) sampling has similarities to DBS. PPS sampling is a multi-stage sampling technique. The data is grouped, and then some subset of the groups are chosen. *Non-uniform sampling* strategies assign varying sampling probability to data so that some specific properties of the original datasets can be better preserved. Non-uniform sampling — best uniform and non-uniform sampling [27, 29]— aims at preserving the relative region density difference. Best uniform sampling preserve the magnitude of density differences while non-uniform sampling increases density differences to alter their magnitude. It divides the sample space into uniform grids and then determines the represented density of each grid, and finally selects samples from each grid according to the density.

*Singular value decomposition (SVD) sampling* is another sampling technique preserving visual density formulated as a matrix decomposition problem and solved with singular value decomposition [138]. This method performs SVD on the original dataset, and selects the samples with the most significant correlation with top-k basis vectors in the SVD result, where $k$ is a rank parameter.
indicating the number of principal components of interest. The SVD based sampling strategy can counterbalance the number of points from regions with different densities.

Recursive subdivision based sampling strategy proposed as multiclass scatterplot sampling includes preserving relative densities, maintaining outliers, and minimizing visual artifacts [50]. It splits the visual space into a binary KD tree and determines which class of instances should be selected at each leaf node based on relative class density by a backtracking procedure. Multi-view Z-order sampling is another density preserving sampling method based on Z-order curve methods [338] and formulated it as a set cover problem. The sets were constructed by segmenting the Z-order curves of the samples in each class and the whole dataset. This strategy selects samples by greedily solving such set cover problems and satisfying results in terms of minimizing kernel density estimation error [128].

The notable feature of this category is maintaining and preserving the relativeness in the visual density of both dense and lighter regions. However, it results in substantial information loss, e.g., Recursive subdivision-based sampling. Also, some of the algorithms are time-intensive, e.g., Multi-view Z-order sampling. These approaches aim to reduce the density in highly occluded regions while preventing a further decrease in the density of sparse areas.

6.3.2.4 Preserving Outliers

Preserving outliers is another general goal in sampling strategies. A typical method for achieving this goal is to update existing sampling algorithms, making them probabilistically accept more outliers according to specified outlier scores [168, 326]. Having no clear definitions, data points in low-density areas are often regarded as outliers [38].
Methods under this category are as follows: *Outlier biased random sampling* assigns higher sampling probabilities to outliers in random sampling [168]. The computed increase in the accepted probability of outliers in the sampling process of blue noise sampling and biased density sampling developed *outlier biased blue noise sampling* and *outlier biased density based sampling*, respectively [168, 326]. *Hashmap based stratified sampling* technique preserve outliers while keeping the main distribution by sampling the point clouds on display using a color mapping [51].

Preserving outliers is in conflict with the intent of preserving relative data densities (high and low-dense area), since more data point selected in low-density regions that distorts relative data densities.

A new method focused on optimizing the point selection in subsampling for overplotted scatterplot using Z-order curves [128]. Space-filling curves create bins, and a seed point from each is selected to represent the set. Z-order for each class and set is calculated for an optimal set cover calculation. This method of solving overdraw provides better class separation. Multi-class scatterplots with heavy overdraw obscure the exploring of outliers, cluster, or local trends, and here hierarchical multi-class technique sampling applies feature-preserving simplification. An interactive-based analysis based on multi-class blue noise sampling method [311] where relative density orders among classes are preserved focused on the optimizing visual abstraction in overdrawn scatterplots [48]. This way, a smooth representation is created, but local patterns are potentially missed. Similarly, Chen et al. applied a non-recursive sampling technique to identify patterns in overdrawn multi-class scatterplots [50]. Their proposed approach better preserve outliers and relative densities. These methods may yield misleading representations, especially in regions where multiple classes are present.
6.3.3 Visual Encoding

Prior studies have demonstrated the effect of visual encodings on analysis tasks [58, 111, 280]. Visual encodings, data and graph aspects influence group or separation perception [320], including color, size, shape [256], orientation [59], texture [12], opacity [186], density [317], motion and animation [49, 94, 296], chart size [124], and others. Additionally, studies have demonstrated a perceptual effect in scatterplots with change in the factors, including data distribution types, number of points, the proximity of concentrations of points, data point opacity, and relative density [56, 67, 109, 111, 147, 242, 280].

The factors, such as, data distribution, number of data points, point size, and point opacity influences the cluster structure on the structure with varying levels of effects and control on the designer [220]. However, data distribution is an inherent property of the dataset and not under the control of designers. On the other hand, the number of data points on the graph can be simulated using the method of sampling. Also, data subsampling solves the challenge of overreading in the scatterplot along with data point size and opacity [92]. These factors directly influence the visual density of the scatterplot, and manipulating these factors means driving the visual density of the scatterplot. As our goal is to improve the effectiveness of the visualization design, it is crucial to understand how designers control factors and use our models to reduce ambiguity in the data. The perception of a cluster is flawed if an overdraw reduction technique does not consider these reasons. Our proposed model explores the visual density on scatterplot using a topology-based model to optimize the design choices in scatterplots.

The focus is on a monochrome scatterplot where raw data are analysed to identify the clusters. From this collection of possible factors, we focus our study specifically on the
factors that most influence visual density, including the number and size of data points, and data point opacity in the visualization. The channels which are not explicitly considered in our study are color and symbol type.

6.3.4 Gaussian Blurring

Data have outliers and noise which biases the interpretation of patterns in a data, e.g., outliers might affect of linear trend in an analysis.

Gaussian blur is a type of image-blurring filter widely used to reduce image noise. The Gaussian function calculates the transformation to apply to each pixel in the image. This is similar to how a mean filter works, but the Gaussian filter uses a different kernel. This kernel is represented with a Gaussian bell-shaped bump. This kernel has some special properties regarding separability that we will look at in detail. The Gaussian distribution in the 1D and 2D cases are shown in equations 6.6 and 6.7, where $\sigma$ is the standard deviation of the distribution.

\[
G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (6.6)
\]

\[
G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (6.7)
\]

Gaussian blurring is another approach similar to kernel density estimation [252] as forms of smoothing represent the image with a color or intensity gradient—these approaches risk distortion from using color to represent a continuum.
Gaussian blurring has its application in the field of visualization. For example, Sato et al. proposed a method for medical volume data that employs a multi-dimensional opacity function to overcome the limitations of 1D opacity functions [249]. Their approach characterized each AOI (tissue) based on its original intensity values and its local intensity structures such as edge, sheet, line, and blob. Their technique employed Gaussian blurring to enhance these specific 3D local intensity structures and use the filter outputs as multi-channel information for tissue classification. Gaussian blurring is also studied in context to splatting to avoid aliasing and produce high image quality without aliasing artifacts or excessive blurring for volume data [340]. Visual complexity on scatterplots displaying multi-dimensional data is enhanced using points in focus and blurry context technique. Staib et al.’s algorithm supports different focus selection bodies, blur kernels, and point shapes and used blur as a visual attribute to explicitly encode out-of-focus distance [273]. Furthermore, using blur allows for extending the influence of the data points forming context in the plot domain and are thus visible behind and between the sharp points.

6.3.5 Threshold Plot

The merge tree-based density threshold is density-based model attempts to directly identify the relative visual density at which users will differentiate between clusters [220]. The density-based model is found by calculating the join tree of a scalar field. I again provide a simplified treatment—see for a detailed description [44]. The Threshold plot, as proposed in chapter 5, is used to calculate the threshold value at which for a given scatterplot X number of clusters could be perceived by the user but does not consider providing the saliency of given clusters on scatterplot designs. Using threshold plots as-is represents an under-constrained optimization, and it requires, at the very least, a user specification of the number of clusters in the data.
Thus models encode the clustering structure as a function of density in the merge tree. The method to select the number of clusters that will be perceived by a user is calculated through the input parameters (density) or the image itself. For this, I generate a persistence threshold plot. For a given merge tree, each component has its threshold, \( \rho \), calculated (Figure 6.3). 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).

6.3.5.1 Cluster Structure Saliency

The automatic parameterization in the optimization model leverages the application of threshold plots to optimize the design decisions. The threshold plot employs density-based merge tree model with improved features of obscuring the noise and outliers. I propose that threshold value on threshold plot (Figure 6.3) computed on the visual density of the scatterplot can act as a proxy for the human perception of cluster structure saliency (Figure 6.4). Cluster structure is identified as more salient with less noise and outliers in the data distribution. Also, values of threshold bars are identified in terms of a maximum of all to pick the best salient cluster structure.

6.3.5.2 Cluster Structure Saliency Interface

To demonstrate the optimization model using the saliency threshold plot, I developed a web interface where one can select an optimized scatterplot based on the clear cluster structure saliency. This optimized design will use the parameters, including the number of points (also called sampling rate), point size and point opacity, and sampling methods. I use eight different datasets, which are widely used in previous studies.
Figure 6.2: Demonstration of the user-guided interactive model. This image depicts the steps in an interactive model to get optimal design choices. (A) User selects one of the eight datasets. (B) User selects the minimum and maximum range of the parameters, including sampling rate, point size, point opacity, and cluster count. (C) Based on the selection, the model presents optimal scatterplot design in the order of their saliency value. (D) Each scatterplot represents a set of optimal choice of design parameters. The model shows nine optimal choices for every iteration.

Our Interface model outputs the results in the form ranking of a scatterplot with cluster structure saliency (on scatterplot design). The user selects the dataset and different factors including sampling rates, point size, opacity, and cluster count. The output ranks the scatterplot saliency, decoded as an optimized design choice for the scatterplot design. The results are encoded as the optimized values of scatterplot: point size value, opacity, sampling rate, and subsampling method. The overview of the interface technique is demonstrated in the Figure 6.1 and working of the interactive model from Figure 6.2.

The flow of threshold plot calculation is demonstrated in Figure 6.3, visual density histogram (B) is calculated on the input image (A). The threshold value (C) is assigned to every cell of
Figure 6.3: Demonstration of threshold plot generation using the visual density. Using scatterplot image and with the density-based model, (A) given the input data of the blurred scatterplot image, (B) a density histogram is calculated. (C) The space is analyzed at different density values, and components are extracted. (D) Tracking components across density values leads to the creation of the merge tree, which marks topological events with nodes. (E) The threshold plots for the merge-tree based density component is created. The horizontal axis represents the threshold, while the vertical axis shows the number of clusters. The red line shows how a threshold can be extracted from a given number of clusters and vice versa.

the histogram using the visual density value. Next, using merge-tree data structure on the threshold value, I calculated the value at which clusters blend to be perceived as one. Finally, I have a graph of threshold values against the cluster count (E).

6.3.5.3 Threshold Plot Bar as Cluster Structure Saliency

A threshold plot step bar length acts as the saliency value (Figure 6.3:E) for a given cluster count. I utilize this to further identify salient cluster structure (Figure 6.4). The salient cluster structure can be identified from the range of cluster count value as our objective here is to select the optimal design choice for a scatterplot that provide the clear cluster structure. As demonstrated in Figure 6.4, there are three prominent threshold bars, and extended bar from the three represents the salient cluster structure. The example from Figure 6.5 shows two scatterplots, and using the threshold plot; we can identify the one with a clear cluster structure, which leads to an optimized scatterplot design choice.
Figure 6.4: The threshold plot for cluster structure saliency. Horizontal axis represents the threshold, while the vertical axis shows the number of clusters. The range bar shows cluster saliency ranking at a given threshold. The salient cluster structure can be identified from the range of clusters. As demonstrated, there are three prominent threshold bars, but the bar \( \textit{3rd} \) represents the salient cluster structure.

As we can see, all three bars here have different length values, and we can label them as high, medium, and low salient based on the bar length (cluster structure saliency). To further incorporate this technique, I plotted a histogram (Figure 6.7(a)) for saliency value for each dataset and performed the binning of high, medium, and low saliency distribution. For the given example of MNIST demonstrated in the Figure 6.7(a), the binning is as 0.0 - 0.33 (L), 0.34 - 0.66 (M), > 0.66 (H).
6.3.5.4 Threshold Plot AUC Difference as Perceptual Similarity

Two similar scatterplot potentially represents similar cluster structure and it becomes ambiguous to distinguish between these two. In this model, we can identify two scatterplot as perceptually similar if their threshold plot similar to each other. To calculate whether two scatterplots have perceptually similar cluster structure, I used a measure called area under the curve (AUC) difference (Figure 6.6), which measure and sum the difference between the threshold plots at each cluster value location. The equation is:

\[ \ell_1(X, Y) = \sum_{i=1}^{n} |x_i - y_i| \]  

(6.8)

Area under curve (AUC) difference value is applied to decide the level of similarity between two scatterplot cluster structures. To further incorporate this technique, I plotted a histogram (Figure 6.7(b)) of the AUC value for each dataset and performed the binning of the AUC distribution into three bins. For the given example of MNIST demonstrated in the Figure 6.7(b), the binning is as 0.0 - 1.1 (Bin 1), 1.2 - 2.5 (Bin 2), > 2.5 (Bin 3). Thus, I have three types of similarity criteria: if the two scatterplots are from the same bin— similar (SR); if the two scatterplots are from the adjacent bin, e.g., 1 and 2 or 2 and 3— somewhat similar (SS); and if the two scatterplots are from bin 1 and 3— dissimilar (DS).

6.4 Implementation of User-Guided Optimization Tool

Next, I develop an interactive tool to demonstrate how to optimize visual encodings and data aspects on real-world data. This work on the top of overdrawn scatterplot solution use different sampling methods contributes to scatterplot designs using sampling methods, point size, and point
Figure 6.5: Cluster structure ranking using threshold plot. Using the threshold value ranking within the range of clusters, one can identify the scatterplot with clearer cluster structure. As demonstrated in the above, for the given range, we have two prominent threshold bars , but the bar represents the clear cluster structure.

Figure 6.6: The threshold plots showing the similar patterns are perceptually similar.

opacity; represents better saliency for clearer cluster structure the cluster structure saliency value calculated on the design factors using a threshold plot.

Here, I detail the stages illustrated in the model overview from Figure 6.1.
6.4.1 Datasets

I selected datasets from the previous studies in visualization as our experiment data to assure the model results' reliability. Since most of them are high-dimensional data, I first transformed them into 2D data using t-SNE and normalized them to $[0,1] \times [0,1]$. I selected eight representative datasets with different characteristics: six datasets with points located as clusters — MNIST [159], Conditional Based Maintenance [63], Clothes [325], Crowdsourced Mapping [137], Epileptic Seizure [11], Swiss Roll 2D [255], and two with curved stripes—Swiss Roll 3D [255], Abalone [82].

6.4.2 Projection

This stage elaborates how the scatterplot images are rendered from 2-D dataset input along with reduction of overplotting.
• Subsampling: Initially, I sub-sampled the dataset using 14 different sampling algorithms

Random sampling, Density biased sampling, Non-uniform sampling, SVD based sampling,
Multi-view Z-order sampling, Recursive subdivision-based sampling, Outlier biased density-based
sampling, Outlier biased random sampling, Hashmap based sampling, Outlier biased blue noise
sampling, Multi-class blue noise sampling, Blue noise sampling, Farthest point sampling, and
Z-order sampling— under 4 categories (section 6.3.2). Each of these algorithms preserves one
or many of the features from datasets— visual density, relative class density, outliers [334]. To better understand the model performance, I selected sampling rates between 0.05 - 0.95
with an interval of 0.05 using given sampling techniques from Table 6.1. Each dataset have 19
sets of sub-sampled files. I observe from studies that number of data points influences visual
tasks such as averaging the values in scatterplot [109], or cluster count in scatterplot [220].
Sampling rate (SR) varies the number of points on the screen.

• Encoding: Data are presented as point marks (i.e., circles) on the scatter plot and two important
visual encodings used in the study are point size (PS) and point opacity (OP). Following design
choices values are selected for the model: point size [20_{px}, 40_{px}, 60_{px}, 80_{px}], and opacity[1%,
5%, 10%, 20%, 40%, 80%]. The reasoning behind choosing values are based on the user study
design from chapter 5.

• Rendering Scatterplots are rendered using the combination of $SR \times PS \times OP$ for all datasets.

  Next, I selected the following visual properties values.

  - Image dimensions ($[X \times Y]$): [700_{px} \times 700_{px}] — The vertical size was selected such that
    the image would fit on the majority of desktop monitors without scrolling [274]. The
    horizontal resolution was selected to match, avoiding any directional bias.
– Data point size/area (PS): \{20_{px}, 40_{px}, 60_{px}, 80_{px}\} — Value for point size was determined analytically.

– Data point opacity (OP): \{1\%, 5\%, 10\%, 20\%, 40\%, 80\%\}.

– Sampling rate (Number of Data Points) (SR): Between the range of \{5\% - 95\%\} with intervals of \{5\%\}.

• Blurring: The noise in the data obscures cluster structure and separation between the clusters, and I proposed blurring to reduce that noise. Instead of calculating the threshold plot on the visual density on the raw rendered scatterplot, I employed Gaussian blurring to hide the noise and outliers. Finally, blurred images are input into the model to provide a saliency threshold plot.

6.4.3 Feature Extraction

The fundamental unit in the model is the visual density which is identified as the feature here. I borrowed data structure from Topological Data Analysis called the merge tree to build models on the visual density. This model attempts to directly identify the relative visual density at which users will differentiate between clusters structure, which is based on join tree of a scalar field—for a detailed description, see [44]. These models try to capture the human perception of clusters by considering the visual density of data points at which cluster distributions will blend to be perceived as one.

To visualize the clustering structure generated by the models, I use threshold plots (Figure 6.3) which plots visual saliency against the number of clusters visible. The threshold plot demonstrates the amount of saliency for the given scatterplot design based on the factors sampling algorithms,
sampling rate (number of points), point size, and opacity of points. I identified the maximum threshold value from all scatterplot thresholds in our work and considered that as the optimized design. Please refer Figure 6.3.

6.4.3.1 Optimized Design

The final results represent the ranked order of scatterplot design based on cluster structure saliency. One point to be noted here is that most scatterplot designs provide similar saliency values because they are identical and, in some cases, perceptually similar (refer to user study). I aim to provide the method to choose the best salient scatterplot design or combination of factors for a given number or scatterplot or a given dataset.

6.5 Computation Time Analysis

The model employs 14 subsampling algorithms on eight heavily overdrawn datasets where execution time varies. To analyze the operational, time-related, and computation information, I experimented with analytical and computation analysis.

6.5.1 Computation Time for Data Subsampling

The selected eight representative datasets have different number of data points as low as 4177 to 70000 on upper limit —MNIST [70000], Conditional Based Maintenance [n=10000], Clothes [n=26569], Crowdsourced Mapping [n=10845], Epileptic Seizure [n=11500], Swiss Roll 2D [n=8000], Swiss Roll 3D [n=10000], Abalone [n=4177]. Some of the sampling algorithms, e.g., blue noise sampling, are time-intensive to sub-sample the dataset. To analyze further, I recorded subsampling time completion for every dataset (8) on the different algorithms (14) at a sampling
rates from 5% - 95% at an interval of 5. I observed a similar completion time for all datasets. Therefore, I will discuss the results for MNIST only.

Figure 6.8 shows the MNIST dataset subsampling completion time plotted for 14 algorithms. We can categorize the subsampling performance based on the completion time into low (Random sampling, Outlier biased random sampling), medium (SVD based sampling, Farthest point sampling), and higher (Blue noise sampling, Outlier biased blue noise sampling). The second important observation here is that some algorithms, e.g., non-uniform, outlier-biased density-based, and random sampling, perform uniformly for all sampling rates. In contrast, algorithms such as blue noise and outlier-biased blue noise completion times increase significantly with the sampling rate.

6.5.2 Computation Time for Projection and Feature Extraction

After the subsampling of the data, we have steps—rendering, blurring, and feature extraction using threshold plot, and these steps are time intensive. In this section, I computed and recorded the time taken for each dataset in terms of the entire set. Entire set comprises of 14 (sampling
algorithms) X 19 (sampling rates) X 4 (data point size values) X 5 (data point opacity values. The
time computation is recorded per scatterplot image from a combination of the set. As we see in the
Figure 6.9, the computed time is proportional to the number of data points in each dataset.

![Figure 6.9: Time analysis for projection and feature extraction. Eight datasets used in the study.]

6.5.3 Scalability

The maximum number of points in the dataset list is 70000 for MNIST. To further analyze
the computation for larger data points in terms of scalability, I selected a dataset—BitcoinHeist [4]
having approximately 3 million data points. I computed and recorded the computation time for
projection, feature extraction as shown in the Figure 6.1. As we can see, the trend in Figure 6.10
demonstrating linear characterization of computation time. The model’s scalability is dependent on
the dataset and the number of points and is linear.

6.6 User Study

To validate the results from the model and further investigate the performance based on the
datasets, I developed a user study on Amazon Mechanical Turk.
6.6.1 Study Design

6.6.1.1 Hypotheses

[H1] More extended the threshold bar, the more salient clusters structure are.

Threshold plots can be used as a proxy to identify which scatterplot designs have more salient structure. A threshold plot bar value is computed on scatterplot visual density which further dependent on design variables: sampling algorithm and rate, point size and opacity in our case. The performance of saliency of clusters can be demonstrated using the bar length.

[H2] Similar patterned threshold plots represent scatterplots that are perceptually similar and have similar cluster structure.

Scatterplots with similar threshold plots shape have similar visual density and visual separation. Hence, visually and perceptually comparing the cluster saliency in the scatterplots with similar threshold plots would be challenging and ambiguous.
6.6.1.2 Stimulus Generation and Selection

- Factor Selection: I selected six datasets from the listed eight datasets. The two were not included in the user study because; Crowdsourced Mapping was used for the training examples for user study, and the second one, Abalone, has similar shape to Swiss roll 3D. Also, varying factors such as sampling technique, rate, point size, or opacity values have minimal effect on the Abalone dataset.

- Scatterplot Rendering: The scatterplot images are rendered with the similar parameters as that are in the interface model.

- Stimuli dimensions ([X × Y]): [700px × 700px] — The vertical size was selected such that the image would fit on the majority of desktop monitors without scrolling [274]. The horizontal resolution was selected to match, avoiding any directional bias.

- Data point size/area (PS): \{20px, 40px, 60px, 80px\}

- Data point opacity (OP): \{1%, 5%, 10%, 20%, 40%, 80%\}.

- Sampling rate (Number of Data Points) (SR): Between the range of \{5% − 95\%\} with interval of 5%.

6.6.1.3 Study Task

Based on the hypotheses H1 and H2, I developed two stimulus tasks for the user study.

[T1] Which of the shown scatterplot have clearer cluster structure?

Two scatterplot designs are shown to the user, and they have to select the one with a clearer cluster structure. Each of scatterplot is assigned a saliency value from our model, and
scatterplot designs with a higher saliency value should have a clear cluster structure. Based on
the selection, the user response is assigned 1 (higher saliency) or 0 (otherwise). I propose that
the user selects the scatterplot with higher saliency, proving that the threshold plot can act as
the proxy for human perception.

[T2] Which of the scatterplots has more similar cluster structure to the reference scatterplot?

A reference and two other scatterplot designs are shown to the user, and they have
to select the scatterplot design with a more similar cluster structure to the reference plot. I
propose that scatterplot with similar threshold step trends have perceptually similar cluster
structures. Based on the selection, the user response is assigned 1 (similar threshold steps) or
0 (otherwise). Scatterplots with similar threshold plots have similar visual density and visual
separation.

6.6.2 Study Procedure

6.6.2.1 Stimulus

Six sets of datasets (D) are selected. Two task sets of T1 and T2 are decided. Three bins (B)
of saliency, High (H), Low (L), and Medium (M) as explained in section 6.3.5.3 are defined for task
T1. Three bins (B) of perceptual similarity, Bin 1, Bin 2, and Bin 3 as explained in section 6.3.5.4
are defined for task T2. Also, there is reference (R) scatterplot for task T2. The stimulus is selected
from the generated pool in random order but in the following manner:

For T1, I showed users two scatterplots as the options and based on Figure 6.7(a) saliency
value is divided into three bins. Next, I have these six combinations for the T1: \( H \times H, H \times M, \)
Each task will randomly select two combinations for each dataset: D(6) × B(2) = 12 stimuli for T1.

For T2, I showed users three scatterplots: one as reference (R) and two as the choice (A or B). Based on the Figure 6.7(b) and as explained in section 6.3.5.4, the AUC is divided into three bins similar (SR), somewhat similar (SS), and dissimilar (DS) based upon from which bins the two other scatterplots are selected. The options are as:

- Option A is similar to R. Option B is dissimilar to R;
- Option A is similar to R. Option B is somewhat similar to R;
- Option A is somewhat similar to R. Option B is dissimilar to R.

Each task will randomly select one combination for each dataset: D(6) × B(1) = 6 stimuli for T2. I intend to keep the number of tasks minimum to reduce the learning effect on the data pattern, and response (in a later stage) should not be biased.

6.6.2.2 Webpage

I developed a webpage for the experiments, where each participant was given 18 (12 × T1 and 6 × T2) stimuli selected randomly from the generated pool. Responses were recorded using selection of scatterplot design (A/B). The maximum allocated time for Task T1 was 10, and T2 was 15 seconds. At the expiration of time, the page was automatically advanced. At the beginning of the experiment, I included a brief introduction, examples and one training task per task type for each participant, which were similar to the study tasks that followed. The experiment was expected to last 30 minutes, including demographic details and training tasks. I also included a post-test
questionnaire, asking participants to describe their criteria for selecting a particular scatterplot design.

### 6.6.2.3 Participants

I recruited participants from Amazon’s Mechanical Turk (AMT) for the IRB approved study [33, 68]. Based upon a post hoc power analysis of the preliminary experiment data, I 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.

In total, for task T1 12 trials $\times$ 70 participants $= 840$ responses, and task T2: 6 trials $\times$ 70 participants $= 420$ responses, were collected. Data quality checks are carried out on responses on these constraints—participants response with timing less than 1 second for a given stimulus task should be rejected; stimulus task with no response or with expired time were rejected. One record for task T1 with no response from the user is identified and hence rejected leaving a total of 839 responses for analysis.

### 6.6.3 Analysis Methodology

#### 6.6.3.1 Task 1

I analyzed data for T1 using a mixed-effects binomial model for the log-odds of a person choosing the higher-salience scatterplot as clearer in trial $t$. I included the threshold bar lengths for the upper-salience (correct) and lower-salience (alternative) scatterplots as fixed-effect predictors and random intercepts for each respondent ($i$) and dataset ($j$).
6.6.3.2 Task 2

I analyzed data for T2 using a mixed-effects binomial model for the log-odds of a person choosing the scatterplot with smaller AUC difference from the reference as more similar in trial $t$. I included the the AUCs for the more-similar (correct) and less-similar (alternative) scatterplots as fixed-effect predictors and random intercepts for each respondent ($i$) and dataset ($j$).

$$\text{ChooseUpper}_{ijt} \sim \text{Binomial}(\lambda_{ijt})$$

$$\text{logit}(\lambda_{ijt}) = \beta_1 \times \text{UpperLength}_{ijt} + \beta_2 \times \text{LowerLength}_{ijt} + \alpha + \gamma_i + \delta_j$$

$$\gamma_i \sim \text{Normal}(\mu_\gamma, \sigma_\gamma)$$

$$\delta_j \sim \text{Normal}(\mu_\delta, \sigma_\delta)$$

(6.9)

$$\text{ChooseMoreSimilar}_{ijt} \sim \text{Binomial}(\lambda_{ijt})$$

$$\text{logit}(\lambda_{ijt}) = \beta_1 \times \text{MoreSimilarAUC}_{ijt} + \beta_2 \times \text{LessSimilarAUC}_{ijt} + \alpha + \gamma_i + \delta_j$$

$$\gamma_i \sim \text{Normal}(\mu_\gamma, \sigma_\gamma)$$

$$\delta_j \sim \text{Normal}(\mu_\delta, \sigma_\delta)$$

(6.10)
6.6.3.3 Analysis Software

I fit models using the glmmTMB package [40, v. 1.1.1] in R [222, v. 4.1.0]. I computed and formatted model results using the modelbased [178] and parameters [173, 174] packages. I managed data using the dplyr [313] and readr [314] packages. I visualized model results using the see [171], ggdist [141], ggplot2 [312], ggtext [315], and patchwork packages [208].

6.6.4 Results

- Task 1: Results for T1 are shown in Table 6.2, Table 6.3, and Figure 6.11. Threshold bar length strongly affected participants’ choice of which scatterplot was clearer. Participants were much more likely to choose the target scatterplot as clearer when there was a large difference in threshold bar lengths between the two scatterplots. This effect was particularly pronounced when the value for the target scatterplot threshold bar length was high.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>log(OR)</th>
<th>SE</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>.71</td>
<td>.26</td>
<td>[0.20, 1.23]</td>
</tr>
<tr>
<td>Target length</td>
<td>.27</td>
<td>.10</td>
<td>[0.07, 0.48]</td>
</tr>
<tr>
<td>Comparison length</td>
<td>-.69</td>
<td>.19</td>
<td>[-1.07, -0.31]</td>
</tr>
<tr>
<td>SD (Intercept: Participant)</td>
<td>.29</td>
<td></td>
<td>[0.09, 0.94]</td>
</tr>
<tr>
<td>SD (Intercept: Dataset)</td>
<td>.23</td>
<td></td>
<td>[0.08, 0.64]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target length</th>
<th>Comparison length</th>
<th>Length difference</th>
<th>Pr (ChooseTarget)</th>
<th>SE</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low [0.15]</td>
<td>Low [0.15]</td>
<td>0.00</td>
<td>0.66</td>
<td>0.06</td>
<td>[0.54, 0.76]</td>
</tr>
<tr>
<td>Medium [0.50]</td>
<td>Medium [0.50]</td>
<td>0.00</td>
<td>0.62</td>
<td>0.06</td>
<td>[0.51, 0.73]</td>
</tr>
<tr>
<td>High [1.50]</td>
<td>High [1.50]</td>
<td>0.00</td>
<td>0.52</td>
<td>0.07</td>
<td>[0.38, 0.66]</td>
</tr>
<tr>
<td>Medium [0.50]</td>
<td>Low [0.15]</td>
<td>0.35</td>
<td>0.68</td>
<td>0.05</td>
<td>[0.56, 0.77]</td>
</tr>
<tr>
<td>High [1.50]</td>
<td>Medium [0.50]</td>
<td>1.00</td>
<td>0.68</td>
<td>0.05</td>
<td>[0.57, 0.78]</td>
</tr>
<tr>
<td>High [1.50]</td>
<td>Low [0.15]</td>
<td>1.35</td>
<td>0.73</td>
<td>0.05</td>
<td>[0.62, 0.82]</td>
</tr>
</tbody>
</table>
Task 2: Results for T2 are shown in Table 6.4, Table 6.5, and Figure 6.12. AUC differences strongly affected participants’ choice of which scatterplot was more similar to the reference scatterplot. Participants were much more likely to choose the target scatterplot as more similar to the reference scatterplot when there was a large difference in AUC values between the two scatterplots. This effect did not substantially vary across target scatterplot AUC levels.

Table 6.4: Model parameters for task 2

<table>
<thead>
<tr>
<th>Parameter</th>
<th>log(OR)</th>
<th>SE</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>.46</td>
<td>.37</td>
<td>[−0.27, 1.19]</td>
</tr>
<tr>
<td>Target AUC</td>
<td>−.51</td>
<td>.29</td>
<td>[−1.09, 0.07]</td>
</tr>
<tr>
<td>Comparison AUC</td>
<td>.64</td>
<td>.18</td>
<td>[0.27, 1.00]</td>
</tr>
<tr>
<td>SD (Intercept: Participant)</td>
<td>.36</td>
<td></td>
<td>[0.12, 1.10]</td>
</tr>
<tr>
<td>SD (Intercept: Dataset)</td>
<td>.17</td>
<td></td>
<td>[0.02, 1.29]</td>
</tr>
</tbody>
</table>
Table 6.5: Model predictions for task 2

<table>
<thead>
<tr>
<th>Target AUC</th>
<th>Comparison AUC</th>
<th>AUC difference</th>
<th>Pr(ChooseTarget)</th>
<th>SE</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR [0.0]</td>
<td>SR [0.0]</td>
<td>0.00</td>
<td>0.61</td>
<td>0.09</td>
<td>[0.43, 0.77]</td>
</tr>
<tr>
<td>SS [1.0]</td>
<td>SS [1.0]</td>
<td>0.00</td>
<td>0.64</td>
<td>0.08</td>
<td>[0.48, 0.78]</td>
</tr>
<tr>
<td>DS [2.0]</td>
<td>DS [2.0]</td>
<td>0.00</td>
<td>0.67</td>
<td>0.11</td>
<td>[0.44, 0.84]</td>
</tr>
<tr>
<td>SR [0.0]</td>
<td>SS [1.0]</td>
<td>−1.00</td>
<td>0.75</td>
<td>0.06</td>
<td>[0.62, 0.85]</td>
</tr>
<tr>
<td>SS [1.0]</td>
<td>DS [2.0]</td>
<td>−1.00</td>
<td>0.77</td>
<td>0.05</td>
<td>[0.65, 0.86]</td>
</tr>
<tr>
<td>DS [2.0]</td>
<td>DS [3.0]</td>
<td>−1.00</td>
<td>0.79</td>
<td>0.08</td>
<td>[0.60, 0.91]</td>
</tr>
<tr>
<td>SR [0.0]</td>
<td>DS [2.0]</td>
<td>−2.00</td>
<td>0.85</td>
<td>0.05</td>
<td>[0.74, 0.92]</td>
</tr>
<tr>
<td>SS [1.0]</td>
<td>DS [3.0]</td>
<td>−2.00</td>
<td>0.86</td>
<td>0.04</td>
<td>[0.75, 0.93]</td>
</tr>
<tr>
<td>DS [2.0]</td>
<td>DS [4.0]</td>
<td>−2.00</td>
<td>0.88</td>
<td>0.06</td>
<td>[0.71, 0.96]</td>
</tr>
<tr>
<td>SR [0.0]</td>
<td>DS [3.0]</td>
<td>−3.00</td>
<td>0.91</td>
<td>0.04</td>
<td>[0.81, 0.96]</td>
</tr>
<tr>
<td>SS [1.0]</td>
<td>DS [4.0]</td>
<td>−3.00</td>
<td>0.92</td>
<td>0.04</td>
<td>[0.82, 0.97]</td>
</tr>
<tr>
<td>SR [0.0]</td>
<td>DS [4.0]</td>
<td>−4.00</td>
<td>0.95</td>
<td>0.03</td>
<td>[0.85, 0.99]</td>
</tr>
</tbody>
</table>

Pr(ChooseTarget) by AUC Difference from Reference

(A) Main model predictions: Darker line indicates smaller target plot AUC

(B) Predictions and 95% CI from model with AUC difference as the predictor

Points are observed binary choice values.

Figure 6.12: User study analysis for task 2.
6.7 Case Study

To evaluate the quality of results and utility of the interface and parameterization model, I conducted a case study with two participants on user-guided automatic parameterization and manual parameterization interfaces.

I recruited two graduate students as voluntary participants from my research lab who are experienced in visualization. To compare the utility of the interface developed and perform qualitative analysis, I developed a manual parameterization interface. I used the same 8 datasets in the following manner: P1 [automatic X 3 and manual X 3; 2 X training] and P2 [automatic X 3 and manual X 3; 2 X training]. The sets for automatic and manual are swapped for both participants.

The objective for each task (for given datasets) is to use the interface (automatic and manual) and select the factors to choose an optimal scatterplot design. The optimal scatterplot represents salient cluster structure. The study is conducted in three parts; part 1: instruction and training, part 2: selecting optimal scatterplot design, and part 3: interview. The total time for each participant was approximately 1 hour.

Based on the study and interview questions, I found both the participants preferred our user-guided parameterization model in terms of interaction, time, and efficiency. As one can observe in Figure 6.13 the manual optimization requires a significantly higher number of interactions (Interaction is defined as selecting the values of the factors for each iteration) to choose a result. However, in terms of time the difference was not significant. The time engagement in the automatic model was caused due to the presence of nine-ranked optimal designs. Finally, I also observed that results chosen by participants using automatic models match the salience bars.
6.8 Discussion

The developed interface model is a data-driven framework to compare and observe how data patterns are high/less sensitive to point size and opacity visual encodings. The data clarity of users’ task performance also depends on the dataset features such as sampling rate or the number of data points.

Hence, I demonstrate this model in ranking the scatterplot design on the saliency of cluster structure based on the number of points (sampling rate), point size, and opacity and sampling methods. The output from the interactive model is a set of scatterplots in the descending order of ranking, representing the cluster saliency value. Saliency values decide the ranking of the scatterplot design for a better cluster structure. Each saliency value encodes features of sampling rate, sampling algorithm, and visual encoding of point size and opacity. An interactive version of optimization is available as OptimizationInterface at <http://scatter.projects.jadorno.com/>.
1. Threshold Plot as a Proxy of Cluster Structure Saliency: As the goal of this chapter 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. Theoretical models of Sadahiro states proximity, and number and concentration, and density change affect cluster perception [242]. Also, the choice of visual factors which influence the visual density of scatterplot can have on small to significant effect on cluster identification [220]. The threshold plot is computed on the visual density estimate of the scatterplots. Threshold bars length, which converts the saliency of cluster count at the given visual density, is extended to find the clearer cluster structure. The longest threshold bars identified from threshold plot generated from scatterplot on the given set of factors; identified as the more salient cluster structure. Further, I validate the results from the user study and participant’s response matched with model-generated results. Based on the findings, we can state that the longest bar in the threshold plot can be used as cluster structure saliency. Visualization practitioners may use the model for optimizing properties of their scatterplot visualizations on different tasks.

2. Perceptually Similar Scatterplots and Cluster Saliency: Every dataset has an inherent property that influences the visualization of the given data. For example, data distribution plays a vital role in points concentration concerning over-saturation or sparse distribution. Such properties often influence the visualization, leading to an ambiguous conclusion of the data even when applying the optimal design choice. I observed and demonstrated in the model that threshold plots (Figure 6.5) that have similar steps/structure and closer in space represent the
perceptually similar scatterplot and cluster structure. Identifying the salient cluster structure is always ambiguous (e.g., for Clothes and Epileptic Seizure datasets) for these scatterplots, and optimizing the design configuration has negligible effects on the visualization. The same has been demonstrated with user study H2 and Task T2, where participants chose the perceptually similar scatterplot with an accuracy of 95.

Different sampling algorithms (see Table 6.1) preserves one or more properties, such as relative visual density, which furthers influence cluster structure separation. However, which sampling algorithm we apply for some datasets with inherent data distribution properties has minimal influences on user perception.

3. Optimal from Good: The proposed user-guided optimization interface provides nine scatterplot configurations demonstrating the cluster structure saliency. All the scatterplot configurations have attached saliency values, and the optimal choice of a given design is choosing best from the suitable combinations.

4. How are they on Computation Time?: Data subsampling is the first step in the process. Algorithms that preserve the spatial separation (e.g., Blue Noise and Outlier-biased blue noise) between clusters take longer to compute as the sampling rate. In contrast, some algorithm performs with the same completion time for all sampling rate. A recent sampling algorithm evaluation study findings say random sampling is preferred for preserving region density; blue noise sampling and random sampling performance well in maintaining relative visual density; outlier biased density-based sampling, recursive subdivision-based sampling, and blue noise sampling perform the best in keeping outliers; and blue noise sampling outperforms the others in maintaining the overall shape of a scatterplot, and spatial separation [334].
5. Next, as observed in the Figure 6.9 majority of datasets required similar time (mean, standard deviation) for the stage of projection and feature extraction. The only outlier is MNIST datasets with a difference of 15 seconds, and the reason is the number of data points for this dataset. We can state the computation time for this stage linearly increases with the number of data points (or sampling rate) as demonstrated in the Figure 6.10. I computed the steps on datasets with higher data points (n= 3 million) and replicated the same up to 9 million to observe a linear pattern.

6.9 Conclusion

Scatterplots are among the most powerful and most widely used techniques for visual data exploration of 2D data. Studies have focused on how scatterplot designs can be optimized and suggested guidelines to render effective visualizations, e.g., visual encodings: data point size and data aspects such as the number of data points. Design choices in visualization, scatterplots in this case, such as the graphical encodings or data aspects, can directly impact the quality of decision making for low-level tasks such as clustering. Effective visualizations improve understanding of data by leveraging visual perception as well the model based on those understanding. Hence, constructing frameworks that consider both the perceptions of the visual encodings and the task being performed enables optimizing visualization to maximize efficacy. However, we were missing a framework that provides an optimized design framework for effective cluster perception in scatterplots using visual encoding and data aspects. I propose here user-guided automatic tools to optimize the design factors of scatterplot for salient cluster structure. Our interactive tool leverages the application of merge tree data structure to optimize the design decisions on — sampling rate, sampling algorithms, symbol
size, and opacity. I further validate our results with a user study and demonstrate the guidelines that practitioners and designers can extend to other tasks on scatterplots.
Chapter 7: Analytical Framework on Effectiveness of Line Charts Smoothing Techniques

7.1 Introduction

Line charts, which date back to William Playfair [215], are commonly used for visualizing time-series and continuous data. Borkin et al. found that line charts are the second most frequently used visualization type, only behind bar charts, in scientific publications, news media, government, and world organizations materials [34].

When line chart data are noisy, e.g., see Figure 7.1(a), visualization designers can turn to smoothing to reduce the visual clutter. However, there are many techniques available (see Figure 7.1(c-f)), and while the results they produce may look similar, each preserves different properties of the data. For example, rank-based (see Figure 7.1(c)) and convolutional smoothing methods (see Figure 7.1(d)) preserve local properties, such as local trends, while frequency-domain smoothing (see Figure 7.1(e)) and subsampling (see Figure 7.1(f)) preserve global properties, such as the most prominent peaks in the data.

To preserve some properties of the input data, each smoothing technique must also lose information, which can have a negative impact on the utility of the resulting data. To further complicate matters, the importance of the lost information can be influenced by both the data being used and the visual analytics tasks being performed. To date, the visualization community lacks a
Figure 7.1: Representation of line smoothing framework. Example of the (a) EEG Channel 10 (500 samples) dataset with (c-f) 4 classes / 12 techniques of line chart smoothing applied. (b) Our framework provides the capability to rank smoothing methods by their efficacy in visual analytics tasks. To form the rankings, the 12 smoothing examples are calibrated to have similar visual complexity. Then, 8 measures of effectiveness, described in section 7.4.1, are calculated and ordered from best to worst. The results show that for this example, the GAUSSIAN, TOPOLOGY, and SAVITZKY-GOLAY methods are the 3 best techniques for the Retrieve Value (RV), Determine Range (DR), Characterize Distribution (CD), Cluster Trends (CT), Sort (S), and Cluster Points (CP) tasks. For the Compute Derived Value (CDV) task, CUTOFF, SAVITZKY-GOLAY, and TOPOLOGY perform best, respectively. Finally, for the Find Extrema (FE) and Find Anamolies (FA) tasks, UNIFORM subsampling, DOUGLAS-PEUCKER, and SAVITZKY-GOLAY perform best. Clearly, no single method is best for all visual analytics tasks, but for this particular data, SAVITZKY-GOLAY, being in the top 3 for all tasks, would be a reasoned choice.
Through this analysis, I show that there is no single smoothing technique that is ideal for all visual analytics tasks. Furthermore, I show that the efficacy of each technique can vary by the datasets being analyzed. Nevertheless, I identify specific methods that consistently perform well, in particular Gaussian filters and topology-based subsampling [236]. In other cases, some methods are particularly well suited for specific tasks, e.g., low-pass cutoff filters and Douglas-Peucker subsampling [80] are well suited for computing a derived value and finding extrema, respectively. Finally, I identify several methods, including the commonly used uniform subsampling, which perform consistently poorly.

Visualization designers can use this framework and results of this chapter to either: (1) select a smoothing technique, which is most effective in general or most effective for the tasks their users perform; (2) evaluate their data to select the technique that is specifically most effective; or (3) to understand how much error is introduced as they increase the level of smoothing used in their visualizations. An interactive version of our framework is at <https://usfdatavisualization.github.io/Lin eSmoothDemo>.

7.2 Prior Work: Line Chart Smoothing

Line charts, which are traditionally used to visualize time-series and continuous 1D data [154, 287, 335], have been studied in the context of a variety of low-level visual analytics tasks [10]. A recent multi-chart experimental study found that line charts are significantly more accurate than other charts for the tasks of correlation and, to a lesser extent, finding extrema, characterizing distributions, and filtering [244]. Even so, line charts are used for a wider variety of visual analytics tasks. I discuss
prior work in the context of analytical tasks performed using line charts, decision-making with line charts in the context of visualization effectiveness in chapter 2.

Smoothing line charts can be considered a form of distortion, as the data are being distorted to improve clarity. There has been prior work looking at smoothing in the signal processing community, e.g., Shao et al. compared 5 smoothing methods for vegetation classification [260], and image processing community, e.g., Chen and Yeh developed a quantitative evaluation for edge-preservation in image smoothing [53]. To a surprise, I was unable to find any prior studies that evaluated the impact of various smoothing techniques to line charts, except for our own small-scale study that introduced a topology-based smoothing method [236]. Nevertheless, no comprehensive framework and evaluation, such as the one I introduced in this chapter, exists.

7.3 Taxonomy of Line Chart Smoothing Approaches

I discuss 4 classes of smoothing that can be used on line charts. They can be broadly broken down into methods that consider local neighborhoods of data or global structures when determining the output. A summary of the 12 smoothing techniques analyzed in this chapter can be found in Table 7.1. Each technique preserves some particular properties of the input through 1 or more adjustable simplification parameters. The number of available smoothing techniques is large. Therefore, this list is intended to be representative of well-known techniques, not necessarily comprehensive.
7.3.1 Local Methods

Local methods only consider nearby data when calculating their smoothed output. Essentially, for each output data, a local neighborhood of the input data is extracted. Then, the neighborhood is processed by a filter, and the result is used as the output.

7.3.1.1 Rank Filters

Rank filters are nonlinear filters that, for each input point, ranks (i.e., sorts) a neighborhood window surrounding the input point. A single value is selected from the ranked set for output. The median filter (see Figure 7.1(c)(left)) selects the median value from the ranked neighborhood. The level of smoothing can be increased or decreased by enlarging or shrinking the neighborhood window, respectively. Median filters are known for being particularly good at removing salt-and-pepper noise [14], but if, on the other hand, those peaks represent important data, they will be lost with a median filter.

To compute the median filter (see Figure 7.2(a)), for each of $n$ input points, a window of size $w$ is first selected. Next, the window is sorted. Finally, the median value is selected for output. The boundary of the domain requires special consideration. Several options exist for the boundary—we chose to repeat the boundary value infinitely. In a naive implementation of the median filter, repeated sorting operations are required, 1 per input/output point, making the overall performance $O(n \cdot w \log w)$. The operation can be optimized by using a sliding window to achieve $O(n \log w)$ in the general case [108] and $O(n)$ in limited cases [212].
Figure 7.2: Illustration of local and global smoothing methods for line charts. Representation of (a-b) local and (c-d) global smoothing methods for line charts. (a) Starting with the input data (top), the \textit{median} filter extracts a window (2nd row), sorts the window (3rd row), and selects the median value for output (4th row). (b) The \textit{Gaussian} filter similarly extracts a local window (2nd row). However, the window has a convolution applied based upon a normal distribution (3rd row), which is used for output (4th row). (c) The low-pass \textit{cutoff} filter converts data into the frequency domain (2nd row), zeros out high-frequency components (3rd row), and converts the smoothed result back into the spatial domain (4th row). (d) The \textit{Douglas-Peucker} method subsamples the input data by iteratively selecting (2nd and 3rd rows) the points with the largest error to insert into the smoothed output (4th row). All techniques are colored by type (see Figure 7.1).

Additional examples of rank filters include \textit{min} filter (see (middle)) and \textit{max} filter (see (right)), which operate similarly, except that they select the minimum and maximum value from the ranked lists, respectively.

7.3.1.2 \textit{Convolutional Filters}

Convolutional filters are a stencil-based method, where for a given input point, a series of weights are applied to a neighborhood surrounding that point. To compute a convolutional filter
(see Figure 7.2(c)), for each of $n$ input points, a window of size $w$ is selected. Next, the elements are multiplied by their corresponding elements from the stencil, summed, and that value is placed in the output. Similar to rank filters, the boundary of the domain requires special consideration. For consistency, I chose to repeat the boundary values infinitely. The resulting computational complexity for general convolutional filters is $O(n \cdot w)$.

The Gaussian filter (see Figure 7.1(d)(left)) is commonly used in convolutional signal and image processing [149]. It weights the input neighborhood using a normal distribution. The smoothing level is increased or decreased by adjusting the standard deviation, $\sigma$, of the distribution. The Gaussian filter can be seen as a form of a low-pass filter, blurring both signal and noise from the data, producing smooth, visually appealing results. The window used for the Gaussian filter is fixed using $\sigma$ as a guide. In our implementation, a window size of $\pm 4\sigma$ ensures that I capture over 99.9% of the distribution.

Another simple convolutional filter is the mean filter (see Figure 7.1(d)(right)), also known as the moving average. In this case, equal weights are applied to all elements in the window, resulting in the average being calculated. Because of the equal weighting, a sliding window can be used to improve performance to $O(n)$ complexity.

Finally, Savitzky-Golay [250] (see Figure 7.1(d)(middle)) is a convolutional filter that uses a low-degree polynomial to smooth the data.

7.3.2 Global Methods

With global methods, the entire input data is considered in the calculation of the output.
7.3.2.1 Frequency Domain Filters

Frequency domain filtering converts the scalar data into a frequency domain representation, via wavelets or Fourier transform. Once in the frequency domain, undesirable frequencies are removed, and the signal is reconstructed. I consider a low-pass cutoff filter (see Figure 7.1(e)(left) and Figure 7.2(b)), which converts the input into the frequency domain using a Discrete Fourier Transform (DFT) [62]. High-frequency components are then zeroed out to smooth the output above a cutoff frequency. Lowering that cutoff frequency increases the level of smoothing. Finally, the output is computed by converting the frequency domain data back to the spatial domain using an inverse DFT. Much like the Gaussian filter, the cutoff filter produces smooth, visually appealing output, in this case, only retaining the specified frequencies. However, the relationship between the frequency and spatial domains is often not intuitive, as multiple frequencies contribute to a single output. The computational complexity of the DFT and the cutoff filter is $O(n \log n)$.

Table 7.1: Summary of smoothing algorithms analyzed.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Class</th>
<th>Average Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEDIAN</td>
<td>Rank</td>
<td>$O(n \log w)$</td>
</tr>
<tr>
<td>MIN</td>
<td>Rank</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>MAX</td>
<td>Rank</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>GAUSSIAN</td>
<td>Convolutional</td>
<td>$O(n \cdot \sigma)$</td>
</tr>
<tr>
<td>MEAN</td>
<td>Convolutional</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>SAVITZKY-GOLAY</td>
<td>Convolutional</td>
<td>$O(n \cdot w)$</td>
</tr>
<tr>
<td>CUTOFF</td>
<td>Frequency</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>BUTTERWORTH</td>
<td>Frequency</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>CHEBYSHEV</td>
<td>Frequency</td>
<td>$O(n \log n)$</td>
</tr>
<tr>
<td>UNIFORM</td>
<td>Subsampling</td>
<td>$O(n + s)$</td>
</tr>
<tr>
<td>DOUGLAS-PEUCKER</td>
<td>Subsampling</td>
<td>$O(s \log n)$</td>
</tr>
<tr>
<td>TOPOLOGY</td>
<td>Subsampling</td>
<td>$O(n + c \log c)$</td>
</tr>
</tbody>
</table>

$n$: number of input points; $w$: window size; $\sigma$: standard deviation of a normal distribution; $s$: number of output samples; $c$: number of critical points (i.e., local minimum or maximum)
Additional frequency domain low-pass filters I consider include the Butterworth filter [41] (see Figure 7.1(e)(middle)) and Chebyshev filter [231] (see Figure 7.1(e)(right)). The cutoff filter is an idealized function that cannot be implemented in an electronic circuit, while the Butterworth and Chebyshev filters can. Practically speaking, these methods differ from the cutoff filter in that they provide a gradual ramp-down of the cutoff frequency.

### 7.3.2.2 Subsampling

Subsampling approaches take the original data and select a subset of the original data points as representatives of the whole data. Simplification is increased by merely selecting fewer points.

A common choice, due to its ease of implementation, uniform subsampling (see Figure 7.1(f)(left)) selects points at regular intervals. Between selected points, interpolation is used, with linear interpolation being the most straightforward case. Uniform subsampling makes few guarantees about the types of features it preserves unless the input is already oversampled, in which case it retains the original signal [259]. Computationally, uniform subsampling is very efficient, only $O(n + s)$, where $s$ is the number of samples taken from the input.

Nonuniform subsampling, in contrast to uniform subsampling, selects points at irregular intervals by considering/preserving some features of the data. Douglas-Peucker [80, 224] (see Figure 7.1(f)(middle)) is an example that establishes a priority queue of points by optimizing the $L^\infty$-norm of the residual error (i.e., the difference between the original and smoothed line charts). The algorithm (see Figure 7.2(d)) starts by selecting the boundary points of the input data (i.e., first and last points) for initialization and connects them via linear interpolation. Points are then iteratively added by selecting the input point with the largest distance from the current output.
and inserting it into the output. The process continues until a user-specified threshold distance is reached. The simplification is increased or decreased by modifying this threshold. The output captured by DOUGLAS-PEUCKER is reliable and predictable, in that the output will deviate no more than the specified threshold. The worst-case complexity of the algorithm is $O(n^2)$, while the average complexity is $O(n \log n)$.

An additional nonuniform subsampling approach, the TOPOLOGY filter (Figure 7.1(f)(right)), uses techniques from Topological Data Analysis to smooth data in a way that retains significant peaks and minimizes error [236]. The TOPOLOGY filter works by first identifying critical points, in the form of local minima and local maxima, and forms a hierarchical pairing (1 each, a local minimum and local maximum) between them. Pairs of critical points are then removed from the output if the difference in their value is below a given simplification threshold. Finally, monotonic regression is used to interpolate between the remaining critical points. The overall complexity of the operation is $O(n + c \log c)$, where $c$ is the number of local minima and maxima.

7.4 Analytical Framework for Measuring for Smoothing Efficacy in Line Charts

In this section, I describe our analytical framework for evaluating line chart smoothing. It consists of 3 parts: a set of effectiveness measures for line chart smoothing (see section 7.4.1); a description of the relationship between the effectiveness measures and common visual analytics tasks (see section 7.4.2); and a description of the methodology for comparing different line chart smoothing techniques (see section 7.4.3).
Figure 7.3: Illustration of the 5 effectiveness measurement used in developed approach. The
example is based upon the low-pass cutoff filter from Figure 7.2(b). (a) shows the total variation,
where \( \ell_1 \) is the sum of the black brackets, and \( \ell_\infty \) is the value of the largest (see arrow).
(b) measures the change in the area, \( \delta a \), by taking the difference between the sum of the grey bars
and the sum of the pink bars. (c) illustrates the peaks in the data in bold. The \( W_1 \) measures the
total difference between peaks, while \( W_\infty \) measures the largest variation. (d) shows the frequency
domain of the input and smoothed data. \( F \) measures the \( L^2 \)-norm of the difference between these 2
vectors. (e) illustrates the (left) value preservation, \( \rho \), and (right) order preservation, \( r_s \), of the
smoothed line chart. Proximity to the diagonal indicates how well the value/order is preserved.

7.4.1 Measures of Effectiveness

To better understand the quality of smoothing results produced by each smoothing technique,
I consider a set of measures that compare the input data, \( X = \{x_0, x_1, x_i, \ldots, x_n\} \), and the smoothed
data, \( Y = \{y_0, y_1, y_i, \ldots, y_n\} \). There is no single measure to evaluate the effectiveness of smoothing
under all visual analytics tasks. Therefore, I use a series of measures, each of which relates how well
the smoothing technique preserves a particular quality of the input data. For all measures, a value
of 0 indicated no error, while larger positive values indicate increasing errors.

7.4.1.1 Total/Maximum Value Variation

The first measures consider calculating the difference between the input and the smoothed
data using vector norms, which measure and sum the difference between the data at each sample
location. Considering the illustration in Figure 7.3(a), I apply 2 variations, the $L^1$-norm and the $L^\infty$-norm.

The $L^1$-norm, $\ell_1$, also known as the least absolute deviations or least absolute errors, measures the sum of the absolute value of the difference between the input and smoothed data. In other words, in Figure 7.3(a), it measures the sum of the differences in black. As a measure, it is robust, in that it is resistant to the influence of outliers. The $L^1$-norm is:

$$\ell_1(X,Y) = \sum_{i=1}^{n} |x_i - y_i|$$  \hfill (7.1)

The $L^\infty$-norm, $\ell_\infty$, measures only the point of the largest difference between input and smoothed data. In Figure 7.3(a), this is the point denoted by the arrow. The $L^\infty$-norm is:

$$\ell_\infty(X,Y) = \max_i |x_i - y_i|$$  \hfill (7.2)

7.4.1.2 Area Preservation

In some cases, the individual deviations matter less than the total area captured under the line chart. The change in the area, $\delta a$, is found by taking the difference between the integrals of the input and smoothed data. Figure 7.3(b) illustrates the process. The change in area is the difference between the sum of all grey bars and the sum of all pink bars. The change in area is:

$$\delta a(X,Y) = \left| \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} y_i \right|$$  \hfill (7.3)
7.4.1.3 Total/Maximum Peak Variation

The next measure identifies and matches the similarity of peaks, i.e., local minima and maxima, between the original and smoothed data. Figure 7.3(c) shows examples of such peaks. To measure the similarity, I use techniques from Topological Data Analysis [86]. First, the local minima and maxima of the original and smoothed data are calculated and paired in a process described in detail in [236]. The pairs are placed into 2 sets $\mathcal{X}$ and $\mathcal{Y}$, and let $\eta$ be a bijection between the two sets.

The Wasserstein distance measures the total difference between all peaks, giving higher weight to those with larger differences. The 1-Wasserstein distance, $W_1$, is:

$$W_1(\mathcal{X}, \mathcal{Y}) = \inf_{\eta: \mathcal{X} \rightarrow \mathcal{Y}} \sum_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \eta(\mathbf{x})\|_1$$

(7.4)

The Bottleneck distance only measures the peaks with the maximum difference. The Bottleneck distance is:

$$W_\infty(\mathcal{X}, \mathcal{Y}) = \inf_{\eta: \mathcal{X} \rightarrow \mathcal{Y}} \sup_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \eta(\mathbf{x})\|_\infty$$

(7.5)

7.4.1.4 Frequency Preservation

Generally speaking, a smoothed signal should maintain as much of the frequency spectrum as possible. To measure the preservation of frequencies, $F$, I convert the original and smoothed data into the frequency domain using the Discrete Fourier Transform (DFT), $F_X$ and $F_Y$, respectively.

\footnote{For technical reasons, all diagonal points ($\mathbf{x}, \mathbf{x}$) are added to make the cardinality infinite [145].}
Once the DFTs are calculated, their difference is found using the $L^2$-norm between them:

$$F(F_X, F_Y) = \sqrt{\sum_{k=1}^{n} (F_{X,k} - F_{Y,k})^2},$$

(7.6)

where $k$ is a single frequency of interest. Figure 7.3(d) illustrates the frequency domain before and after smoothing. The frequency preservation would be the $L^2$-norm of the difference between these 2 vectors.

### 7.4.1.5 Value-Order Preservation

In some scenarios, knowing that the relative values of data items are maintained is more important than maintaining the correct values. The value-order relationship can be measured using the correlation between the input and smoothed data. To measure the relationship between relative values, the Pearson Correlation Coefficient, $\rho$, can be employed. Figure 7.3(e)(left) illustrates the value relationship by placing points at $(x_i, y_i)$. Intuitively, the $\rho$ measures the proximity of the points to the diagonal in grey. In order to treat $\rho$ consistently with other measures, I modify it, such that 0 is a perfect positive correlation, and 2 is a perfect negative correlation. The modified $\rho$ is:

$$\rho(X,Y) = 1 - \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$$

(7.7)

The order relationship between data items can be measured using Spearman Rank Correlation, $r_s$, which is the Pearson Correlation Coefficient of the ranked data, in other words:

$$r_s(X,Y) = \rho(rank(X), rank(Y))$$

(7.8)
Figure 7.3(e)(right) illustrates the order relationship. The points are placed by rank, instead of value, and their $\rho$ is calculated.

### 7.4.2 Low-Level Task Taxonomy for Line Charts

To determine relevant visual analytics tasks, I adapt the low-level task taxonomy of Amar et al. [10] to line charts. For each task, I provide a brief description and an example query on the line chart in Figure 7.4. Finally, I relate each of these tasks to 1 or 2 of the metrics from the previous section, which is summarized in Table 7.2.

#### 7.4.2.1 Retrieve Value

This task is focused on finding the function value at an exact location in a given dataset/chart (Figure 7.5(a)). For example, using Figure 7.4, “What is the stock price on Mar ’15?” (answer: $\sim 4.5$). The accuracy of retrieving a value is dependent upon how closely chart values match the data value, which mirrors the total/maximum value variation measures (section 7.4.1.1 and Figure 7.3(a)). Measuring the total difference between the smoothed data and the original data, in other words,
Table 7.2: Matrix of tasks and metrics

<table>
<thead>
<tr>
<th>Retrieve Value</th>
<th>✓</th>
<th>✓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determine Range</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Compute Derived Value</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Find Extrema</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Find Anomalies</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Characterize Distribution</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Sort</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cluster Trends</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Points</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

the $L^1$-norm (Equation 7.1), provides an average case performance for the retrieve value task. By measuring the maximum difference between the smoothed and original function, in other words, the $L^\infty$-norm (Equation 7.2), the worst case performance can be calculated.

7.4.2.2 Determine Range

For this task, specific criteria, e.g., a range of values, are provided for identifying data points, e.g., the dates/times that are within that range (Figure 7.5(b)). Using Figure 7.4, an example is, “What months saw values between 3 and 4?” (answer: Oct ’14, Nov ’14, Feb ’15, Mar ’15). Accuracy in performing this task is highly dependent upon the criteria provided. Nevertheless, generally, it is important that the values of the data closely reflect those of the input data, in other words, the
total/maximum value variation (section 7.4.1.1 and Figure 7.3(a)). The *average case* is measured using the $L^1$-norm (Equation 7.1). The *worst case* is measured using the $L^\infty$-norm (Equation 7.2).

7.4.2.3 Compute Derived Value

This task focuses on computing an aggregate, such as the average or total value of a function (see Figure 7.6(a)). For example, using Figure 7.4, “What is the average stock price from Oct ’14 to Apr ’15?” (answer: $\sim 3$). The task accuracy is mostly dependent on how well the line chart globally (i.e., the sum of all values) matches the input data. The task essentially requires the user to visually determine an integral of the data, which is equivalent to the area preservation measure (see section 7.4.1.2 and Figure 7.3(b)). The area preservation measure, $\delta a$, provides the *average case* performance by measuring the difference in integrals between the original and smoothed data.

7.4.2.4 Find Extrema

This task is concerned with finding local minima and maxima (i.e., valleys and peaks) in the data (see Figure 7.6(b)). For example, using Figure 7.4, “What are the dates/values of all of the peaks in the data?” (answer: Nov ’14/4.5, Jan ’15/3, Mar ’15/5). The accuracy of the task depends upon the peaks and/or valleys remaining present in the output and significant enough to be
visible. The total/maximum peak variation (see section 7.4.1.3 and Figure 7.3(c)) measures both the existence and significance of peaks in the data. The *average case* is provided by the 1-Wasserstein distance (Equation 7.4), which finds the total variation in peaks between the input and smoothed data. The *worst case* is found using the Bottleneck distance (Equation 7.5), which measures the peak of maximal variation.

### 7.4.2.5 Find Anomalies

This task involves looking for values that do not conform to the overall trend in the data (Figure 7.7(a)). For example, “Between Nov ’14 and Feb ’15, what month, if any, does not follow the data trend?” (answer: Dec ’14/Jan ’15). The task of finding anomalies is similar to finding extrema, in that anomalies are generally peaks in the data, but in this case, they do not follow the trend of the data. Since the task involves identifying peaks, the *average case* is provided by the 1-Wasserstein distance (Equation 7.4), and the *worst case* is found using the Bottleneck distance (Equation 7.5). Interestingly, the removal of anomalies is also one of the reasons smoothing is applied to line charts. Therefore, when performing other tasks, the *preservation of anomalies might be considered a negative quality.*
7.4.2.6 Characterize Distribution

This task involves summarizing a trend in the data (Figure 7.7(b)). For example, using Figure 7.4, “What is the trend in the data between Nov ’14 and Feb ’15?” (answer: downward). Trends in the data are synonymous with the frequency domain of the data. To be effective, the frequency domain of the smoothed data should be as similar as possible to that of the input data. Therefore, the average case accuracy of this task is measurable using the frequency preservation measure (section 7.4.1.4 and Figure 7.3(d)) from Equation 7.6.

7.4.2.7 Sort

This task asks users to, given some criteria, rank or order the data values (Figure 7.8(a)). An example, using Figure 7.4, would be, “What is the order of stock values for the dates Nov ’14, Jan ’15, and Mar ’15, from lowest to highest?” (answer: Jan ’15, Nov ’14, Mar ’15). While similar to the retrieving value task, the accuracy of this task relies both upon the relative order of values (not the exact values) of data remaining the same, and further, the difference between those relative values is reasonably discernible. The value-order preservation measures (section 7.4.1.5 and Figure 7.3(e)) provide 2 mechanisms to understand the average case performance. First, Spearman Rank Correlation ( Equation 7.8) can be used to compare the relative order of all points in the
original and smoothed data. The Pearson Correlation Coefficient (Equation 7.7) can be used to determine, on average, how discernible the values in the smoothed data are from one another, as compared to the input.

7.4.2.8 Cluster

This task asks users to group data with similar values or trends (Figure 7.8(b)). For example, “Group months with similar trends” (answer: left). Depending upon the nature of the query (clustering individual points vs. clustering trends), this task depends upon both the judgment of relative values and trends in the data. Therefore, the average case performance is summarized by the frequency preservation measure (section 7.4.1.4 and Figure 7.3(d)) from Equation 7.6 for clustering trends, as well as the value-order presentation measures (section 7.4.1.5 and Figure 7.3(e)), Spearman Rank Correlation (Equation 7.8) and Pearson Correlation Coefficient (Equation 7.7) for clustering individual points.

7.4.2.9 Tasks Not Considered

Amar et al. [10] defined additional low-level tasks that I found redundant or out-of-scope for this analysis. First was the filter task, which I felt was redundant with tasks such as determine
range, characterize distribution, and find anomalies. The second was the *correlate* task, which I felt would require comparing multiple distributions for their similarity. While this task could potentially be analyzed with the framework, I only consider the effectiveness of tasks on a single line chart.

7.4.3 Evaluation Framework

Given the metrics and tasks described in the prior subsections, I establish this framework for comparing the efficacy of smoothing techniques. The idea is to rank the effectiveness of all smoothing techniques for given data on a specific visual analytics task.

This requires 2 things: (1) a common measure of the smoothing level (i.e., a baseline for measurement); and (2) a method for ranking the efficacy of methods, using a specific metric.

7.4.3.1 Visual Complexity as a Proxy for Smoothing Level

As noted in this taxonomy of smoothing techniques, each method provides 1 or more input parameters for adjusting the level of smoothing. However, the input parameters for each technique have little to no direct relationship to any other technique. For example, *Gaussian* smoothing with $\sigma = 5$ samples has no analog to *uniform* subsampling 50% of points, even though they may produce similar results. This makes the analytical comparison of techniques difficult.

Recently, approximate entropy (*ApEx*) was shown to be a high quality proxy for the visual complexity of line charts [240]. Generally speaking, approximate entropy is a measure that quantifies predictability of fluctuations in the data. In our case, I see visual complexity and smoothing level as synonymous—therefore, *ApEx* is used as a baseline for our analysis. In other words, if the smoothed outputs of 2 different techniques have the same *ApEx*, I consider their smoothing level identical, e.g.,
in Figure 7.1, all methods have similar ApEx values. See [240] for a description of how to compute ApEx.

7.4.3.2 Ranking the Effectiveness of Smoothing

To select the most effective technique for a given metric, I want to focus on those that have the smallest value.

- Ranking a Single Smoothing Level: When smoothing results have equivalent ApEx, ranking their effectiveness is fairly trivial. For a given metric, the techniques are simply ordered from lowest (best) to highest (worst). For example, in Figure 7.1(b), the $L_1$-norm, $\ell_1$, in the first column, shows that TOPOLOGY has the lowest error, followed by GAUSSIAN and SAVITZKY-GOLAY. The rankings can be computed for all metrics, and the smoothing techniques evaluated for all tasks. For example, in Figure 7.1(b), SAVITZKY-GOLAY is in the top 3 for all metrics/tasks, making it a reasonable choice to represent the input data.

![Figure 7.9](image)

Figure 7.9: Entropy plot for the EEG Channel 10 (500 samples) dataset. TOPOLOGY (in red) and CHEBYSHEV (in purple). The $L_1$-norm is plotted vertically, and the ApEx horizontally. (a) The methods are sampled at 100 levels to capture a range of ApEx values. (b) Linear and logarithmic regression are used to obtain a best-fit model. (c) Mathematical integration is used to capture the area under the curve. Methods with smaller areas produce less error, thus performing better. In this case, TOPOLOGY is $\sim 2500$, while CHEBYSHEV is $\sim 4350$, making TOPOLOGY the more effective method.
Ranking All Smoothing Levels: Summarizing the performance of different smoothing techniques across all smoothing levels requires additional analysis. I calculate 100 different smoothing levels, across a range of entropy values. For each metric, I create an entropy plot, which is a scatterplot of the metric value against a range of ApEx values. In Figure 7.9(a), the $L^1$-norm is plotted vertically, against the ApEx horizontally for topology, in red, and Chebyshev, in purple.

Next, both linear and logarithmic regression are performed using iterative reweighted least-squares (IRLS)\textsuperscript{18} [125]. The model, linear or logarithmic, with the larger $R^2$ value is selected as a proxy for the efficacy of the technique. For Figure 7.9(b), topology is best modeled logarithmically, and Chebyshev is best modeled linearly.

Since the goal is again to minimize the error induced in the data, the total area under the regression is computed (i.e., mathematical integration), and the methods are ranked smallest to largest area. In Figure 7.9(c), the total area is $\sim 2500$ for topology and $\sim 4350$ for Chebyshev, making topology more effective than Chebyshev.

\textsuperscript{18}Using IRLS helps to minimize the impact of outliers.
The resulting ranks are placed into a rank plot, as seen in Figure 7.11. In this plot, the $L^1$-norm is ranked across multiple datasets. Each dataset receives a column, and the smoothing methods are ranked from best (top) to worst (bottom). The tracks are added to improve readability.

- Ranking Across Multiple Datasets: To summarize the overall efficacy of techniques across multiple datasets, an average rank is calculated. The average rank simply takes the sum of the rank across all datasets and orders them from best (top) to worst (bottom). In Figure 7.11, the average rank is the final column. The average rank of \textsc{topology} is \( \frac{4+2+2+1+1+1}{6} = 1.8 \), while the \textsc{Gaussian} is 2.0, \textsc{Savitzky-Golay} is 3.5, etc.

Figure 7.11: Rank plot of the $L^1$-norm for all EEG (500 samples) datasets. Each column ranks the techniques, from best (top) to worst (bottom), on a different dataset, with the average rank in the final column. The result shows that for this data, on average, \textsc{topology}, \textsc{Gaussian}, and \textsc{Savitzky-Golay} are the most effective techniques, respectively.
7.5 Results

The source code and data for our framework and evaluation is available at <https://github.com/USFDataVisualization/LineSmooth>, and an interactive version of framework is as LineSmooth at <https://usfdatavisualization.github.io/LineSmoothDemo>.

7.5.1 Data Sources

I evaluate 80 datasets in 13 categories from 8 data sources (see Figure 7.10). Data were selected that contained typical qualities, such as long-term trends (e.g., stock prices), cyclical behaviors (e.g., daily temperatures), or important spikes (e.g., a stock market crash).

- Chicago Homicide Rates (chi_homicide) data (see Figure 7.10) contains weekly (969 samples) and monthly (222 samples) counts of the number of homicides in the city from January 2001 through July 2019. Data is provided by the City of Chicago [52].

- EEG (eeg_500, eeg_2500, and eeg_10000) data (see Figure 7.10) contains windows of 3 different lengths (500, 2500, and 10000 samples) from 6 (of 32 total) channels from a single subject undergoing a visual attention task and was acquired from the EEG/ERP Public Archive [70].

- New Zealand Tourist (nz_tourist) data (see Figure 7.10) contains the monthly (1165 samples) and annual (96 samples) number of tourists visiting the country from April 1921 through April 2018. Data collected from Trading Economics [192].
- US Domestic Flights (*flights*) data (see Figure 7.10) contains the number of daily, weekly, and monthly (7671, 1095, and 252 samples, respectively) number of US flights from January 1, 1988 through December 31, 2008. Data collected from Observable [35].

- Stock Price (*stock_price*) and Stock Volume (*stock_volume*) data (see Figure 7.10) contains daily closing values and trading volumes, respectively, for 9 companies (Apple, Amazon, Bank of America, Google, Intel, JP Morgan, Microsoft, Toyota, and Tesla) over a 5 year period, January 2015 through December 2019 (1257 samples each), collected from Yahoo Finance [328].

- Average Wind Speed (*climate_awnd*), High Temperature (*climate_tmax*), and Total Precipitation (*climate_prcp*) data (see Figure 7.10) contains 10 years of daily weather values (3651 samples each) from 6 US Airports (Atlanta, New York JFK, Los Angeles, Chicago O’Hare, Seattle-Tacoma, and Salt Lake City), collected from NOAA Climate Data Service [190].

- US Unemployment (*unemployment*) data (see Figure 7.10) are the monthly number of unemployed individuals in 14 economic sectors (e.g., agriculture, finance, health, etc.) from January 2000 through February 2010 (122 samples each). The data were collected from the US Bureau of Labor Statistics [291].

- Radio Astronomy (*astro*) data (see Figure 7.10) contains 5 spectral “lines” (1947 samples each) that measure the frequency and amplitude of radio waves emitted by extraterrestrial matter (i.e., gas and dust), collected from the ALMA Science Archive [9].

### 7.5.2 Evaluation By Task

I evaluate 12 smoothing methods from Table 7.1 with our framework (see Figure 7.1). Tasks that use the same metrics are combined to reduce space. I produce rank plots summarizing all
datasets for the metrics related to those tasks. Each column is the average rank for a given data category, with the average rank across all 80 datasets (i.e., total performance) in the final column. To reduce the clutter, I only show the tracks for smoothing methods with the top 4 overall performance.

7.5.2.1 Retrieve Value / Determine Range

The results for Retrieve Value and Determine Range tasks, in Figure 7.12, show that 3 smoothing techniques—TOPOLOGY, GAUSSIAN, and SAVITZKY-GOLAY—produced the best results. Still among those datasets, there is a distinction between results depending upon the dataset category. For example, TOPOLOGY excelled at data that are predominated by “spiky” features, e.g., \textit{climate_prcp} and \textit{stock_volume}. On the other hand, GAUSSIAN and SAVITZKY-GOLAY performed better on data with long-term trends, e.g., \textit{stock_price}, and cyclical behaviors, e.g., \textit{climate_tmax}. Among the worst performing techniques were all rank-based approaches, all frequency domain-based approaches, and the UNIFORM and DOUGLAS-PUECKER subsampling approaches.

7.5.2.2 Compute Derived Value

The results for the Computing Derived Value task, in Figure 7.13, show that the \textit{CUTOFF} filter clearly outperforms all other techniques. Examining the entropy plots (not shown), it can be observed that TOPOLOGY performs similarly well on most of the data sets. Finally, the convolutional methods, GAUSSIAN, \textit{MEAN}, and SAVITZKY-GOLAY, occasionally performed well. Among the worst performing techniques are all rank-based techniques, frequency domain-based techniques, excluding \textit{CUTOFF}, and subsampling techniques, excluding TOPOLOGY.
Figure 7.12: Average ranking for the Retrieve Value and Determine Range tasks. The ranking is across all datasets and uses (a) $L^1$-norm and (b) $L^\infty$-norm. Both metrics show that TOPOLOGY and GAUSSIAN were primarily the best methods to use, while SAVITZKY-GOLAY occasionally performs well.

7.5.2.3 Find Extrema / Find Anomalies

The results for Find Extrema and Find Anomalies tasks, in Figure 7.14, show that DOUGLAS-Peucker produced the best results, with GAUSSIAN, TOPOLOGY, and, interestingly, UNIFORM occasionally performing well. Since the subsampling techniques, including UNIFORM, use a subset of the original data, it is safe to assume that for some levels of smoothing, peaks will be included in the output. Among the techniques that performed poorly were once again rank-based, frequency domain-based, and convolutional techniques, excluding GAUSSIAN.
Figure 7.13: Average ranking for the Compute Derived Value task. The ranking is across all datasets using the area preservation metric, $\delta a$. The results show that CUTOFF is clearly the top method, followed by TOPOLOGY. GAUSSIAN, MEAN, and SAVITZKY-GOLAY occasionally perform well.

7.5.2.4 Characterize Distribution / Cluster: Trends

The results for the Characterize Distribution and Cluster: Trends task, in Figure 7.15, show that depending upon the datasets, GAUSSIAN, TOPOLOGY, or SAVITZKY-GOLAY produce the best results, with TOPOLOGY working best on “spiky” datasets and convolutional techniques working better on those datasets with long-term trends and cyclical behaviors. Among the worst performing techniques are again all rank-based techniques, frequency domain-based techniques, and subsampling, excluding TOPOLOGY.

7.5.2.5 Sort / Cluster: Points

The results for the Sort and Cluster: Points tasks, in Figure 7.16, show that GAUSSIAN was the best performer, followed by TOPOLOGY and SAVITZKY-GOLAY, depending upon whether the Pearson metric, in Figure 7.16(a), or the Spearman metric, in Figure 7.16(b), is used. Once again, the TOPOLOGY method appears to work best on “spiky” datasets, while the convolutional methods...
worked better on data with long-term trends or cyclical behaviors. The worst performing techniques for these tasks are largely the same as for other tasks.

7.6 Discussion: Smoothing Recommendations

Given our evaluation in section 7.5, I summarize the efficacy of the techniques tested in Table 7.3. For these grades, I measure the frequency of a method being ranked in the top 3 for a given task across each of the 80 datasets. Grades are assigned using that frequency: A: > 75%; B:
Figure 7.15: Average ranking for the Characterize Distribution and Cluster: Trends tasks. The ranking is across all datasets using the frequency preservation metric, $F$. The results show that depending upon the type of data, GAUSSIAN, TOPOLOGY, and SAVITZKY-GOLAY methods produce the best results.

50% − 75%; C: 25% − 50%; D: 5% − 25%. In other words, a method scoring a grade of A ranks 1st, 2nd, or 3rd in at least 75% of datasets.

1. General Recommendation: If designing a visualization without particular concern for the data or visual analytics task, GAUSSIAN and TOPOLOGY performed well in all categories. The main difference between the 2 is that while GAUSSIAN had more A’s, TOPOLOGY has no score lower than B. The heart of which to pick really lies in the type of data being visualized. As I pointed out in our evaluation, TOPOLOGY tended to do better on data with “spiky” features, while GAUSSIAN did better with cyclical behaviors or long-term trends.

2. Task Specific Recommendations: If the visual analytics tasks are known ahead of time, a more nuanced decision can be made about what method to use. While GAUSSIAN and TOPOLOGY did well on most tasks, CUTOFF and DOUGLAS-PEUCKER performed the best on Compute Derived Value and Find Extrema/Anomalies, respectively.
Figure 7.16: Average ranking for the Sort and Cluster: Points tasks. The ranking is across all datasets using the (a) Pearson Correlation Coefficient, $\rho$, and (b) Spearman Rank Correlation, $r_s$. Both metrics show Gaussian performing best, followed by topology and Savitzky-Golay.

3. Data Specific Recommendations: If the data are available ahead of time, analyzing them with our framework to select the best smoothing method is recommended. In addition to Gaussian and topology, several methods generate better results in limited situations, particularly if the tasks are known as well. These methods include median, Savitzky-Golay, mean, cutoff, and Douglas-Peucker.

4. Methods to Largely Avoid: Several methods performed poorly across the board. These include min, max, Butterworth, Chebyshev, and uniform subsampling. These methods rarely performed in the top 3. They should only be used when there is a very specific reason to do so,
Table 7.3: Grades for the efficacy of different smoothing methods. Grades are calculated by the frequency of being ranked in the top 3 for all 80 datasets. A: > 75%; B: 50% – 75%; C: 25% – 50%; D: 5% – 25%

<table>
<thead>
<tr>
<th></th>
<th>Compute</th>
<th>Derived</th>
<th>Range</th>
<th>Find Extrema</th>
<th>Characterize</th>
<th>Distribution</th>
<th>Cluster: Trends</th>
<th>Cluster: Points</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MEDIAN</strong></td>
<td>D</td>
<td>D</td>
<td>–</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td><strong>MIN</strong></td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>MAX</strong></td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>D</td>
<td>D</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>GAUSSIAN</strong></td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>B</td>
<td>B</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td><strong>Savitzky-Golay</strong></td>
<td>C</td>
<td>C</td>
<td>D</td>
<td>D</td>
<td>B</td>
<td>C</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td><strong>MEAN</strong></td>
<td>C</td>
<td>C</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td><strong>CUTOFF</strong></td>
<td>D</td>
<td>D</td>
<td>A</td>
<td>D</td>
<td>D</td>
<td>C</td>
<td>D</td>
<td>C</td>
</tr>
<tr>
<td><strong>Butterworth</strong></td>
<td>–</td>
<td>–</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>Chebyshev</strong></td>
<td>–</td>
<td>–</td>
<td>D</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>Uniform</strong></td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>D</td>
<td>D</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>Douglas-Peucker</strong></td>
<td>D</td>
<td>D</td>
<td>–</td>
<td>A</td>
<td>A</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td><strong>Topology</strong></td>
<td>B</td>
<td>B</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

which should not be a problem as most of these techniques are rarely used anyways. However, this finding is particularly relevant for **uniform** subsampling, as it is essentially the default methodology used for data reduction [117].
7.7 Conclusions

In conclusion, I have presented and demonstrated a framework for evaluating line chart smoothing in the context of the visual analytics tasks being performed. There remain several study limitations and future works.

- Perceptual Effects and User-based Validation: Our study considers only the effects of data modification in the evaluation of smoothing effectiveness. There may be additional perceptual effects that make the results of some techniques better or worse than others. I considered performing a user study to validate our framework further. However, it became quickly apparent that the scale of such a study would be impractical. As an example, testing 12 smoothing techniques, across 8 tasks, 80 datasets, and 20 different smoothing levels would require in excess of 150k experimental stimuli.

- Feature Types and Representing Lost Information: Throughout our analysis, I discussed “spiky”, cyclical, and long-term trends in data. These categories are ill-defined, and a broader study of feature types that appear in line charts would be valuable to the community. Furthermore, smoothing removes information from the representation of the line chart, introducing uncertainty. One additional direction of future work would be to use this framework to model and represent the uncertainty while considering the context of the visual analytics task.

- There’s No Accounting for Taste: Aesthetics play an important role in visualization design. Without a good aesthetic, users are less likely to remember what they see [34]. Although TOPOLOGY and GAUSSIAN were largely the most effective techniques, their aesthetics are quite different, “spiky” for TOPOLOGY and smooth for GAUSSIAN. Our framework completely ignores
aesthetic in its recommendation, in part because aesthetic is both art and science, thus difficult to model mathematically or algorithmically.
8.1 Conclusion

Design optimization is a perennial topic in information visualization, striving to effectively visualize the data for better interpretation and accurate user task performance. The way we visualize the data, whether it is encoding data on the graphs using visual channels, e.g., colors, symbol-types, sizes, etc.; data aspects, e.g., number of data points; or performing analysis tasks, e.g., identifying correlation, can influence the conclusions people draw using data. Visualization designs must consciously consider optimizing these critical elements—visual encoding, data aspects, and low-level tasks. This dissertation defines the need for this intersectional study on visual encoding and tasks to fill the gap using a task-optimized framework. The models demonstrate how designers can produce optimal design choices. The aim is to provide high-level overviews of optimal visual design for better quality and higher confidence in decision-making that gives designers objective guidance.

Visual encodings are fundamental to visual design and have a long history of use in visualization [25]. Graphical perception term introduced by [58] resembles visual encoding suggests that an optimal design is required to estimate precise values. The trend in perceptual studies demonstrated the application of graphical perception and other perception concepts (section 2.1) in visualization design studies. The analytic task’s evaluation on visualization design can be improved using perceptual (e.g., for judgments of relationship with visual channels and accuracy or discrimination) and cognitive (e.g., completion time or error rate) measures on the visual design, focusing on effectiveness.
The perception-based study results evaluate design guidelines on individual visualizations and visual channels. Yet, as the findings around visualization effectiveness are nuanced, guidelines for the effective visualization design depend on the visual channels used in chart types and analysis tasks, which points to a significant need to understand the intersection of these factors to create optimized visualization. One of the important findings from the report states that visualization effectiveness is task-dependent. This survey also illuminates the progress on understanding graphical perception along with areas requiring more study.

Modeling human perception to optimize design decisions based on visual factors (visual encoding, e.g., symbol type or area; or data aspects, e.g., a number of data symbols) is possible through perceptual models, and effectiveness can be improved using perceptual evaluation. I proposed a framework for task-optimized visualization to guide design decisions that amplify the visualization effectiveness and improve data communication and interpretation. The framework provides techniques for and measures the effectiveness of visualization in terms of visual designs and tasks on widely-used visualizations. An effective visualization must be intuitive to make decisions easier to interpret, improve public communication and discussion, and promote public trust in sound science. In this dissertation, I prototype the framework on two visualizations, scatterplots and line charts, focusing on clustering tasks and smoothing line graphs.

Design optimization using visual encoding and data improves user performance on low-level visual tasks, data interpretation, and communication. Design choices in visualization, scatterplots in this dissertation, such as the graphical encodings or data aspects, can directly impact the quality of decision-making for low-level tasks, such as clustering. Using a topology-based model, I investigated and evaluated the influence of visual factors and encoding on cluster count perception. The findings
demonstrated a significant level of influence of the factors. Further, I utilized these findings in a user-guided optimization model that suggests optimal design choices for a scatterplot on cluster saliency.

An optimal design of visualization for visual tasks includes data transformation, e.g., smoothing on noisy line charts. However, there are many smoothing techniques available, and while the results they produce may look similar, each preserves different properties of the data. In this dissertation, I present an analytical framework for measuring the effectiveness of various smoothing techniques under eight other low-level visual analytics tasks performed on line charts. The framework evaluation shows that no single smoothing technique is ideal for all visual analytics tasks. Furthermore, the efficacy of each method can vary by the dataset being analyzed. The framework and results can be extended to select and optimize line charts by selecting a smoothing technique, which is most effective in general or most effective for the tasks their users perform.

These findings collectively inform how to optimize a visualization design on visual encoding, data aspects, and data transformation to support the user’s performance at low-level visual tasks.

8.2 Limitations

There are several limitations in this work. Many provide exciting roads for future work without lessen the contributions of this dissertation.

8.2.1 Design Optimization in Scatterplots and Line Charts

- Study Limitations in Cluster Perception: The study itself has some additional limitations.

First, I have not considered other factors that could influence performance in either model,
e.g., chart size, screen resolutions, etc., and visual encoding that affects the visual density, e.g., symbol type or color. I have also not extensively analyzed variance between individuals, although I did note some slight variation during our analysis (i.e., some individuals had over-or under-estimation tendencies). Another limitation is that I have only provided a limited analysis of mixing effects, e.g., changing the size of points while also changing the opacity. A final limitation is that I have not considered the correlation between confidence, which is highly related to the nature of data [96], and the correctness of each model.

- Study Limitations in Line Smoothing: The current line smoothing framework evaluates the effectiveness of smoothing techniques on a developed analytical framework that uses the measure of effectiveness. This work focuses on the majority of low-level tasks. However, a perceptual evaluation framework will open new research questions on efficacy precision and estimate judgment. Also, a design-based evaluation of line graphs (e.g., line opacity, line stroke, line width, and others) would provide a detailed guideline to an optimized design framework.

8.2.2 Task-optimized Framework

- Supporting Different Visual Low-Level Tasks: This discussion of low-level tasks focuses on specific types, e.g., clustering in scatterplot or most tasks in line smoothing work. Though the framework at this level at the start of the task-optimized framework, a broader consideration of the different low-level tasks on various visualization is necessary for a viable and more general design framework. This will open new research questions for graphical perception and visual tasks and help designers support these tasks in practice.
• Scalability Evaluation in terms of Datasets: The scalability in terms of datasets has been considered in cluster-based scatterplot design optimization (8 datasets with a varied number of records) and line smoothing (13 different datasets), and I found the results to be assuring. However, considering datasets of different types and being heavily overcrowded could potentially validate the limitation of the models. Understanding when and why designers pick a given visual design rather than computational methods is essential for understanding its utility in practice.

• Interaction and System: All of the studies in this dissertation demonstrated the work with some form of interaction providing the working insights of the framework to the user, e.g., TopoCluster for chapter 5, OptimizationInterface for chapter 6, or LineSmooth for chapter 7. While interaction can complicate study design, it provides several potential benefits for completing visualization tasks. For example, instead of manually estimating an optimal design, the user can interact with data and parameters on the interface to compute exact optimized scatterplot design choices. An advanced and more comprehensive interface system for the optimization framework would demonstrate a better understanding of how interaction and user understanding can be used in practice to task framework that would substantially inform visualization design.

8.3 Future Work

This work represents first steps in many directions. Many potential research projects could build on this work.
8.3.1 Consider Other Visualizations

This dissertation focuses on scatterplots and line charts, focusing on clustering and line smoothing, respectively. Many of the knowledge gaps and design guidelines addressed in this dissertation exist for other visualization as well. For example, choropleth maps are well studied for correlation tasks in perception. In visualization, a map plots temporal and spatial information, and lots of time is only considered for a few tasks. A better task-optimized design provides a better understanding of information and practical data interpretation.

8.3.2 Consider Other Visual Encoding

In this work, I consider how design factors influence the visual density of the scatterplot. The visual density value of the display varies with other encodings factors and is less susceptible than encodings studied in this dissertation. I want to focus on colors, symbol types, display aspect ratio and investigate their effect on visual design in the future. For line charts, building a perceptual framework with user-validation on visual encodings such as width, color, mark type would lead to effective and task-optimized visualizations.

8.3.3 Multi-Visualization Optimization System Deployment

The user-guided optimization tool provides the optimal design decisions for scatterplot and clustering tasks. Though this parameterization interface is one-visualization focused, it could be extended to multiple visualizations. Currently, I propose incorporating visual encodings, such as mark types and color, and chart properties, such as aspect ratio, in subsequent work. I hope to increase the value of user-guided systems by making them available as multi-task and multi-visualization.
8.3.4 Improving the Effectiveness of User-Guided Optimization Tool

The analytic task’s evaluation on visualization design can be improved using eye-tracking, think-aloud, and perceptual (e.g., for judgments of 1) relationship with visual channels and accuracy, 2) discrimination using JND), and cognitive (e.g., completion time or error rate) measures on the visual design, focusing on effectiveness. Such measures would explore the more profound understanding of the data interpretation through visualization. An effective visualization must be intuitive to make decisions easier to interpret, improve public communication and discussion, and promote public trust in sound science.

8.3.5 Diverse Participatory Recruitment for User Study Evaluation

Users at different education strata often process and abstract visual information differently, which is traditionally not the focus of the visualization design. Hence, I will recruit a diverse set of domain experts and concerned population groups (expert and non-experts, new to visualization, and less educated users) in the evaluation user study. Integrating design insight from a diverse range of participants will enable us to develop insights into visualization effectiveness that overcome historic inequities in the ways visualization design has been evaluated by encompassing a broader range of experiences. By integrating 1) frameworks established on design decisions, task, and visualization with perceptual evaluation, and 2) inclusive design guidelines, the resulting framework will provide empirically-grounded guidelines for selecting the best technique for visualizing data tasks in mind.

These works collectively reframe how visualization designers and practitioners can think about visual design optimization. They present evidence of the utility of task-optimized methods for reframing how designers reason about two-dimensional visualization optimization. I do not expect
the work displayed here to be the final word in considering visualization designs in practice. I see this dissertation preferably as guiding new conversations about task-optimized visualization.
References


233


239


[222] R Core Team. R: A language and environment for statistical computing.


Appendix A: Copyright Permissions for Chapter 3, 4 and 7

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author’s approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/credit notice should be placed prominently in the references: © [Year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication].
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis online.
3) In placing the thesis on the author’s university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity’s name goes here]’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

© 2021 Copyright - All Rights Reserved | Copyright Clearance Center, Inc. | Privacy statement | Terms and Conditions
Comments? We would like to hear from you. E-mail us at customercare@copyright.com

249
A Survey of Perception-Based Visualization Studies by Task
Author: Ghulam Jilani Quadri
Publication: IEEE Transactions on Visualization and Computer Graphics
Publisher: IEEE
Date: Dec 31, 1969
Copyright © 1969, IEEE

Thesis / Dissertation Reuse
The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:
1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source [author, paper, publication] followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author’s approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:
1) The following IEEE copyright/credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication].
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis online.
3) In placing the thesis on the author’s university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity’s name goes here]’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.
LineSmooth: An Analytical Framework for Evaluating the Effectiveness of Smoothing Techniques on Line Charts

Author: Paul Rosen
Publication: IEEE Transactions on Visualization and Computer Graphics
Publisher: IEEE
Date: Feb. 2021
Copyright © 2021, IEEE

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author’s approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication].
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis online.
3) In placing the thesis on the author’s university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity’s name goes here]’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/publishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.
4) If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

© 2021 Copyright - All Rights Reserved | Copyright Clearance Center, Inc. | Privacy statement | Terms and Conditions
Comments? We would like to hear from you. E-mail us at customercare@copyright.com
Appendix B: Institutional Review Board Authorization

The following is the IRB approval for the cluster perception and design optimization study (chapter 5 and chapter 6).

12/4/2017

Ghulam Jilani Quadri
Computer Science and Engineering
4202 E Fowler Ave
Tampa, FL 33620

RE: Exempt Certification
IRB#: Pro00033201
Title: Feature Identification from Various Visualization Using Mechanical Turk: An Empirical Measure Study

Dear Mr. Quadri:

On 12/4/2017, the Institutional Review Board (IRB) determined that your research meets criteria for exemption from the federal regulations as outlined by 45CFR46.101(b):

(2) Research involving the use of educational tests (cognitive, diagnostic, aptitude, achievement), survey procedures, interview procedures or observation of public behavior, unless:
(i) information obtained is recorded in such a manner that human subjects can be identified, directly or through identifiers linked to the subjects; and (ii) any disclosure of the human subjects' responses outside the research could reasonably place the subjects at risk of criminal or civil liability or be damaging to the subjects' financial standing, employability, or reputation.

As the principal investigator for this study, it is your responsibility to ensure that this research is conducted as outlined in your application and consistent with the ethical principles outlined in the Belmont Report and with USF HRPP policies and procedures.

Please note, as per USF HRPP Policy, once the Exempt determination is made, the application is closed in ARC. Any proposed or anticipated changes to the study design that was previously declared exempt from IRB review must be submitted to the IRB as a new study prior to initiation of the change. However, administrative changes, including changes in research personnel, do not warrant an amendment or new application.

Given the determination of exemption, this application is being closed in ARC. This does not limit your ability to conduct your research project.

We appreciate your dedication to the ethical conduct of human subject research at the University of South Florida and your continued commitment to human research protections. If you have