

ABSTRACT

Title of Dissertation: **SIMULATION, REPRESENTATION, AND
AUTOMATION: HUMAN-CENTERED
ARTIFICIAL INTELLIGENCE FOR
AUGMENTING VISUALIZATION DESIGN**

**Sungbok Shin
Doctor of Philosophy, 2024**

Dissertation Directed by: **Professor Niklas Elmqvist
College of Information Studies,
Department of Computer Science**

Data visualization is a powerful strategy for using graphics to represent data for effective communication and analysis. Unfortunately, creating effective data visualizations is a challenge for both novice and expert design users. The task often involves an iterative process of trial and error, which by its nature, is time-consuming. Designers frequently seek feedback to ensure their visualizations convey the intended message clearly to their target audience. However, obtaining feedback from peers can be challenging, and alternatives like user studies or crowdsourcing is costly and time-consuming. This suggests the potential for a tool that can provide design feedback for visualizations. To that end, I create a virtual, human vision-inspired system that looks into the visualization design and provides feedback on it using various AI techniques. The goal is not to replicate an exact version of a human eye. Instead, my work aims to develop a practical and effective system that delivers design feedback to visualization designers, utilizing advanced

AI techniques, such as deep neural networks (DNNs) and large language models (LLMs).

My thesis includes three distinct works, each aimed at developing a virtual system inspired by human vision using AI techniques. Specifically, these works focus on simulation, representation, and automation, collectively progressing toward the aim. First, I develop a methodology to simulate human perception in machines through a virtual eye tracker named A SCANNER DEEPLY. This involves gathering eye gaze data from chart images and training them using a DNN. Second, I focus on effectively and pragmatically representing a virtual human vision-inspired system by creating PERCEPTUAL PAT, which includes a suite of perceptually-based filters. Third, I automate the feedback generation process with VISUALIZATIONARY, leveraging large language models to enhance the automation. I report on challenges and lessons learned about the key components and design considerations that help visualization designers. Finally, I end the dissertation by discussing future research directions for using AI for augmenting visualization design process.

SIMULATION, REPRESENTATION, AND AUTOMATION:
HUMAN-CENTERED ARTIFICIAL INTELLIGENCE FOR
AUGMENTING VISUALIZATION DESIGN

by

Sungbok Shin

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2024

Advisory Committee:

Professor Niklas Elmqvist, Chair/Advisor
Professor Philip Resnik, Dean's Representative
Professor Zhicheng (Leo) Liu
Professor Christopher A. Metzler
Professor Sanghyun Hong

© Copyright by
Sungbok Shin
2024

Dedication

“The Lord is my shepherd: I shall not want.”

PSALM 23:1

To God, whom I shall not want.

Acknowledgments

I am very proud to have completed my Ph.D. from one of the most prestigious CS programs in the United States, and possibly the World (**Let's go Terps!**). I really worked hard to grow as a researcher. Fortunately, I am blessed with numerous helpers along the way. I wouldn't be where I am today without the remarkable and thoughtful guidance of my advisors and friends. Their knowledge and perspectives have been fundamental in guiding my academic path. For that, I express my sincere gratitude for their contribution and support.

First and foremost, I would like to acknowledge my advisor, **Dr. Niklas Elmqvist**, for his pivotal role in my development as both a researcher and an individual. He is my role model.

Second, I would like to give great thanks to **Dr. Sanghyun Hong**. I will never forget his assistance during my early struggles to establish myself as a researcher. His contribution to my research journey has been very significant.

Third, I must also acknowledge my Master's advisor, **Dr. Jaegul Choo**. His mentorship and support throughout my Master's studies at Korea University have been crucial to my achievements today.

Last but not least, I would like to acknowledge my parents, **Sunsuk Kim and Dongwon Shin**. Their constant support was the key factor that made it possible to overcome the challenging periods of my Ph.D. journey.

Now, I would also like to thank other members of my committee, Profs. Philip Resnik, Christopher Metzler, Zhicheng Leo Liu.

Next, my teammates in VisTeam@HCIL@UMD, who are: Dr. Andrea Batch, soon-to-be-doctors Md Naimul Hoque, Pramod Chundary and Biswaksen Patnaik.

Next, Tom Hurst, Migo Gui, and Jodie Gray from the CS department, who helped me immensely throughout my journey as a CS Ph.D. student, from the start to graduation.

Next, my collaborators, who have helped me during my Ph.D.. They are: Dr. Panagiotis D. Ritsos (Bangor), Dr. Peter W. S. Butcher (Bangor), Sunghyo Chung (CS@KU), Inyoun Na (CS@KU), Julia Liu, and Hyeon Jeon (SNU).

Next, people from the University of Maryland: Dr. Youngjoon Lee from Political Science department, and Dr. Seokbin Kang, Dr. Young-Ho Kim and his wife Hyomin, Dr. Kyungjoon Lee, Dr. Jonggi Hong, Jaehoon Choi, and Geonsun Lee from the CS department.

Next, my friends at CS@KU: Iksoon Yang, Jisik Woo, Jongwook Jeong, Dalsu Choi, Minje Choi, Youngduk Seo, and Wooil Kim.

Finally, I would also like to thank Yohan Cho, Sungwoo Chun, Soobin Lee, and Hyunjoon Park.

Upon graduation, I will be starting my academic career as a postdoc in Team Aviz@Inria, Paris, France with Prof. Jean-Daniel Fekete. As a researcher, I will always try to conduct research that is useful to our community, our society, and ultimately, our World.

Table of Contents

Dedication	ii
Acknowledgements	iii
Table of Contents	iv
List of Tables	viii
List of Figures	x
List of Abbreviations	xiv
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 A Virtual, Human Vision-inspired System for Augmenting Visualization Design	3
1.2.1 Simulation: A Scanner Deeply	3
1.2.2 Representation: Perceptual Pat	4
1.2.3 Automation: Visualizationary	5
1.3 Contributions	5
1.4 Dissertation Structure	7
1.5 Prior Publications and Authorship	7
Chapter 2: Background	9
2.1 Facilitating Visualization Design	9
2.2 Teaming with AI for Visualization Design	10
Chapter 3: A Scanner Deeply: Predicting Gaze Heatmaps on Visualizations Using Crowd-sourced Eye Movement Data	12
3.1 Overview	15
3.1.1 Understanding Visual Perception via Eye Movements	16
3.1.2 Choice of Apparatus and Techniques in Prior Work	17
3.1.3 Our Approach	20
3.2 Task-Specific Crowdsourced Data Collection	22
3.2.1 Visualization Image Collection	23
3.2.2 Gaze Collection Setup	25
3.2.3 Participant Task	26
3.2.4 Collection Results	29

3.3	A Scanner Deeply	29
3.3.1	SimpleNet: A DNN for Gaze Prediction	30
3.3.2	Preprocessing Webcam-based Eyetracking Data	31
3.3.3	Training SimpleNet on Our Crowdsourced Dataset	32
3.4	Evaluation	34
3.4.1	Prediction Performance of the Scanner Deeply	34
3.4.2	How Does the Scanner Deeply Perceive Charts?	36
3.4.3	What Structure Does the Scanner Deeply Focus?	39
3.4.4	Analyzing Temporal Patterns in Human Perception	41
3.5	Discussion	42
3.5.1	Assessing Crowdsourced, Webcam-based Gaze	42
3.5.2	A Priori or A Posteriori?	43
3.5.3	Limitations and Future Work	44
3.6	Conclusion and Future Work	45
Chapter 4: Perceptual Pat: A Virtual Human Visual System for Iterative Visualization Design		
4.1	Background	50
4.1.1	The Role of Feedback in Visualizations	50
4.1.2	Facilitating the Chart Design Process	51
4.1.3	Understanding Human Perception for Visualizations	52
4.1.4	Modeling Human Perception	54
4.2	Design: Virtual Human Visual System	55
4.2.1	Motivation	56
4.2.2	Scope	56
4.2.3	Design Requirements	57
4.2.4	Practical Perceptual Feedback	58
4.3	The Perceptual Pat Suite	64
4.3.1	System Overview	64
4.3.2	The Pat Design Lab	66
4.3.3	Current Perceptual Pat Components	69
4.3.4	Implementation Notes	71
4.4	User Study	71
4.4.1	Participants	73
4.4.2	Apparatus	73
4.4.3	Design Task	73
4.4.4	Procedure	75
4.4.5	Data Collection	77
4.5	Results	77
4.5.1	Results from the Initial Interview	77
4.5.2	Evolution of Visualization Snapshots	80
4.5.3	Evolution Rationale	82
4.5.4	External Assessment	85
4.5.5	Post-experimental Interview	86
4.6	Discussion	89

4.6.1	Benefits of the Perceptual Pat Suite	89
4.6.2	Limitations and Future Work	90
4.7	Conclusion	93
Chapter 5: Visualizationary: Automating Design Feedback for Visualization Designers		
	Using Large Language Models	94
5.1	Background	96
5.1.1	Design Feedback	96
5.1.2	Automatic Design Feedback on Visualizations	97
5.1.3	Large Language Models	99
5.2	Model: Automated Visualization Design Feedback	100
5.2.1	Target Audience	100
5.2.2	Rationale	101
5.2.3	Analyze-Clarify-Guide-Track (ACGT) Workflow	101
5.2.4	Visualization Filters	103
5.3	The Visualizationary Tool	104
5.3.1	System Overview	104
5.3.2	Visualization Design Management Interface	106
5.3.3	Visualization Metrics	107
5.3.4	Visualization Design Reports	110
5.3.5	Revision Tracker	111
5.3.6	Prompt Engineering for Visualization Feedback	111
5.3.7	Implementation Notes	114
5.4	User Study	114
5.4.1	Participants	117
5.4.2	Task	117
5.4.3	Procedure	118
5.4.4	Data Collection	120
5.4.5	Apparatus	120
5.5	Results	120
5.5.1	Design Evolution	121
5.5.2	Post-Study Interview	135
5.5.3	Design Evaluation	138
5.6	Discussion	139
5.6.1	On Visualization Design Feedback	139
5.6.2	Limitations and Future Work	140
5.7	Conclusion	143
Chapter 6: Conclusion		144
6.1	Summary	144
6.2	Future Work	145
Appendix A: Appendix – Visualizationary		147
A.1	Prompts used for Visualizationary	147
A.1.1	Preamble used for Visualizationary	147

A.1.2 Prompts used for Visualizationary using ChatGPT	149
A.2 Evaluation with ChatGPT and Wolfram-GPT	168
Bibliography	172

List of Tables

3.1	Taxonomy of prior work. We categorize the 30 papers that collect eye movement data with three dimensions: (1) the tracker, (2) image stimulus, and (3) collection methodology. Note that the papers are listed in chronological order. We found that there is no work that employs the same collection approach as ours, i.e., collecting webcam-based crowdsourcing eye movements on chart images. . . .	14
3.2	Example queries for collecting chart images from the web. Each query (on the right) is a combination of two keywords from our topical analysis of papers (on the left) and a chart type (in the middle).	23
3.3	Performance evaluation of Scanner Deeply by comparing it with gaze prediction models proposed by prior work. We show the performance using five metrics. We report each metric’s mean and standard deviation over the five runs. The value shown on top is the mean, and on the bracket below is the standard deviation over the five runs. We compare with DVS [1] and Salicon [2]. The only difference between Salicon and the Scanner Deeply is in the dataset. We show that a model using datasets collected for the specific task and domain on a specific task can show improved performance over a general-purpose gaze prediction model. We also compare ours with Scanner Deeply before pre-processing. This is represented as Ours [†]	35
4.1	Components composing Pat’s feedback system. Names, feedback type and implementation of seven components included in the suite. These were selected to be representative and cover the design space in § 4.2.4. We provide additional technical and performance details on these components in Appendix (supplemental material).	68
4.2	User study demographics. Our participants were recruited from the Data Visualization Society Slack channel (https://www.datavisualizationsociety.org/slack-community) and community of IT Engineers in Facebook South Korea. * = participant P5 did not finish the study.	72
4.3	Initial interview with participants. Responses for Q1 are given in the participant’s own order.	79
4.4	External assessments. Average 1-5 Likert scale ratings (1 = significant decline, 3 = neutral, 5 = significant improvement) by external evaluators assessing the quality of changes from one version to another ($v_x \rightarrow v_y$) as well as overall from the initial to the final version. Standard deviations are given within parentheses. .	86

4.5	Post-experimental interview with participants. After each participant finished updating their versions of visualization, in the post-experimental interview we asked 6 following questions to understand how they get feedback and whether they are familiar with a tool that provides design feedback. In Q1, the answers are written in the order each participant mentioned.	87
4.6	Advantages and disadvantages of Perceptual Pat. After the experiment, we asked participants about the advantages/disadvantages of a virtual human visual system providing visual feedback for visualization designers. The table shows a summary of the answers from 4 visualization designers that participated in our experiment.	87
5.1	User study demographics. A total of 13 visualization designers of different seniority participated in the user study. These participants were all above the age of 18, had at least a bachelor’s degree in visualization or related field, and had more than three years of experience in designing visualizations. We categorize those with less than or equal to 3 years of experience in visualization as novice (6 participants), those with 4 to 7 years as intermediate (4 participants), and those with more than 7 years as expert groups (3 participants), respectively.	116
5.2	Questions for post-study interviews. We asked participants a total of 7 questions about their experience in using Visualizationary to understand the role of Visualizationary in the design process, to unveil their decision-making process in acting on feedback, and to evaluate the tool.	134
5.3	Evaluation of design evolution. We asked three external experts to evaluate the improvements in the designs generated by 13 participants. Scores are measured using a 1–5 Likert scale, where 1 represents a significant decline, 3 is neutral, and 5 is a significant improvement. Each value shows the mean value of scores from external evaluators; values in parentheses are standard deviations.	135
A.1	Comparison of averaged feedback scores of participants’ designs between 3 experts, 3 ChatGPT trials, and 3 WolframGPT trials.	168
A.2	An example of design feedback about P7’s works from an expert, ChatGPT, and WolframGPT	170

List of Figures

3.1	Summary of the Scanner Deeply pipeline. Pipeline for developing a gaze prediction model that, given an image as an input, produces a saliency map for an image containing a visualization. First, we collect more than 10,000 images that contain one chart per image (A). Second, using the collected images we conduct a crowdsourced study on Amazon Mechanical Turk to gather gaze logs using a webcam eyetracker (B). Third, using the image collection with gaze log annotations, we train a model called SimpleNet, a CNN-based neural network model (C). Gaze predictions are shown in (D) in the form of saliency maps.	13
3.2	Calibration and gaze collection. Participants click on 8 dots that are placed near the boundaries of the screen. Each dot turns yellow once the user has clicked on it five times (A-1). Then, the calibration accuracy is measured by making users look at the middle dot at the center of the screen (A-2). If the accuracy is above 70%, then participants can proceed to the gaze collection phase (A-3). In the gaze collection phase, participants are asked to look at the chart for 7 seconds, during which time the user’s gaze is tracked (B-1). Then, they are asked to answer what type of chart the image is (B-2). After they submit the answer, participants look at the next image by clicking “Go” button at the question “Ready for the next image?” We customize the code WebGazer developed by Papoutsaki et al. [3] in creating the website.	15
3.3	Chart type popularity. We show the 10 most popular charts from a set of 1,000 samples randomly chosen from our image collection. This provides an estimate on the chart type distribution in our dataset.	24
3.4	Illustration of procedures for constructing oracle heatmaps. We use the final heatmaps for training our neural networks. Gaze dots collected by using webcams contain rich information that enables further analysis, e.g., temporal changes in human visual perception (see §3.4.4 for the analyses), but they are also noisy to learn. We address this challenge by carefully pre-processing gaze dots. The second column shows the dots in the raw data from webcams, and we blur dots and remove some of them not in the human’s area of focus (see §3.3.2 for details).	27
3.5	Our SimpleNet architecture. The pipeline accepts image input and produces an output image. In our case, the input is a visualization and the output is the gaze map. SimpleNet follows an encoder-decoder structure similar to U-Net [4]. Compared to DNNs from prior work [2], SimpleNet requires less computation.	31

3.6	Visualization of gaze predictions. We show the gaze predictions of three charts from ScannerDeeply, DVS, and SimpleNet trained on the Salicon dataset. DVS and Salicon focus more on text, while our model focuses more on distinct areas in a chart.	37
3.7	Gaze predictions from ScannerDeeply for different types of visualizations. We show examples of the six common chart types in our datasets, <i>i.e.</i> , (from the left) a line chart, a bar chart, a population pyramid, a heatmap on a geographic map, a scatterplot on a geographic map, and a scatterplot. We analyze them in §3.4.3.	38
3.8	Temporal eye movements in gaze predictions. By training SimpleNet on datasets representing different time scale, ScannerDeeply can predict human visual perception over time. In (A), we train our model on the gaze maps collected from the first 0.7 seconds, while we train our models on the gaze maps for the entire 7 seconds. Our model predicts that human eyes will focus first on the distinct areas in charts and then focus on text components (e.g., legends) later on.	41
4.1	Pat Design Lab. Applying Perceptual Pat to Charles Minard’s famous Napoleon’s March to Russia diagram. The Pat Design Lab is the web-based interface to the Perceptual Pat suite of perceptually-inspired image filters that can be used by a designer to receive rapid and inexpensive feedback on a visualization being designed.	48
4.2	The Pat Design Lab in action. Example of a designer using the Pat Design Lab during their chart design process.	55
4.3	System overview. The Perceptual Pat suite is comprised of a user interface (The Pat Design Lab) and a feedback system. The user interface receives a data visualization (chart) from a user and automatically performs an analysis of the chart. The feedback system consists of diverse visual analysis components and uses them to generate a report containing design feedback. (We used Perceptual Pat to iteratively refine this figure by providing a freelance graphic designer with the original version and a PDL report.)	64
4.4	Designing visualizations with the Pat Design Lab. The user can log in to the system by signing in on (A). Then, they can upload images on (B), and start the analysis process by clicking the ‘analyze’ button in (C). They can choose filters in the control view, and at the same time control the opacity of the filter in (D). To take notes in the report, they need to click the blue button in (E). Then, they can add texts in (J) and click the button ‘add.’ Users have to manually press the save button in (F) to keep record of them report with notes. They can check her past versions of the design by clicking the hyperlink ‘archive’ at the navigation bar (G). In the archive view, they can select a version of her interest in (H), and the most-recently chosen version will appear at the left window, pushing the old one into the right window (I).	65

4.5	Report generated by the Pat Design Lab. The report contains information about filters used for the analysis of the updated chart images. It consists of 8 sections. These include (1) the chart’s specifications, (2) OCR, (3) visual entropy, (4) a Scanner Deeply, (5) low-level salience, (6) potentially distracting objects (Chartjunk [5]), and (7) color-vision deficiency.	66
4.6	Overview of collected designs. These images represent the five most significant versions (as identified by the participants) during the design process of creating a new visualization artifact from scratch using the Pat Design Lab and the Perceptual Pat Suite. All four participants from the study are represented (five were recruited, but one person abandoned the study after a week).	81
5.1	Interactive report. Report on a visualization artifact based on our novel analyze-clarify-guide-track (ACGT) framework. It is generated by automatically analyzing (“analyze”) the artifact using an extensible collection of image filters grouped into categories: salience, text, visual representation, color, and accessibility. After reporting results from each image filter (center column), the report then interprets (“clarify”) the findings using natural language understandable to a novice visualization designer. This is followed by natural language suggestions (“guide”) on how to address weaknesses in the design. Finally, the right side of the report shows changes over time (“track”) as the designer iteratively refines the artifact. The texts in black boxes are responses generated by ChatGPT.	94
5.2	System overview. Visualizationary is composed of a visualization design management interface and a feedback system; the white circles refer to the analysis and report steps, respectively. The system first receives a data visualization (chart) from the project container interface ① and generates design feedback (explanations ③ and suggestions ④) based on the filters used to analyze ② the visualization. The feedback system is designed to support the <i>analysis-clarify-guide-track</i> workflow and leverages a large language model (LLM) to automate the steps. We archive the feedback generated by Visualizationary ⑤ with the revision tracker. The design report component ⑥ summarizes the feedback and the revision history for the iterative design improvement process.	105
5.3	Visualizationary design management interface. The system shows an overview of the timeline of revisions for a visualization artifact. Each revision consists of a screenshot of the artifact as well as an interactive design report. The right column shows the revision tracker, which provides a high-level view of the overall design trajectory.	106
5.4	Design evolution (P1-P3). Five versions (as identified by the participants) during the design process of creating a new visualization artifact from scratch using Perceptual Pat. The first three participants from the study are represented.	122
5.5	Design evolution (P4-P6). Five versions (as identified by the participants) during the design process of creating a new visualization artifact from scratch using Perceptual Pat. The next three participants from the study are represented.	125
5.6	Design evolution (P7-P10). Five versions (as identified by the participants) during the design process of creating a new visualization artifact from scratch using Perceptual Pat. The last four participants from the study are represented.	128

5.7 **Design evolution (P11-P13).** Five versions (as identified by the participants) during the design process of creating a new visualization artifact from scratch using Visualizationary. The next three participants from the study are represented. 132

List of Abbreviations

AI	Artificial intelligence
VHVS	Virtual Human Visual System
ML	Machine Learning

Chapter 1: Introduction

This is an introducing chapter that talks about the motivation, and brief introduction on each work conducted in this thesis.

1.1 Motivation

I describe the motivation by first talking about the problems in visualization, the potential of AI, and finally, augmenting the visualization design process using AI.

Problems in visualization. Information visualization, according to Ben Shneiderman, is a “*powerful interactive strategy for exploring data, especially when combined with statistical methods.*”

Data visualizations are used every day to obtain quick insights in various businesses and industries. To enhance this effectiveness, there have been various efforts to understand how people read visualizations and how visualization design processes take place. Based on these works, various chart authoring tools have been created, lowering our barrier for creating visualizations and widening our expressiveness when drawing visualizations. In spite of these advancements, however, we still identify issues in creating visualizations. First, people make mistakes when creating visualizations, resulting in distorted visualizations for various purposes. Second, creating effective visualizations is difficult for both novice and expert visualization designers. Novice designers may struggle to use effective methods for a given task, and experts, though knowledge-

able, may still want to assess the effectiveness of their work during their design process. Despite these problems, however, there is not a well-established solution to address these issues, where designers are guided or informed based on their design decisions.

Potential of AI. The rapid advancement of artificial intelligence (AI) is changing various parts of our daily lives. This rapid swift of paradigm is also changing every part of research community, including data visualization. Especially, with the development of versatile large language models (e.g., ChatGPT) and diffusion models (e.g., Midjourney, Stable diffusion), we observe tools that look as if they can ‘answer’ as if they were humans. In visualization community, we observe numerous attempts to use visualization for interpreting AI models. While we are in early stages, we are starting to see more of applications to use AI for enhancing authoring/reading of visualizations, such as in creating linting techniques, or generating captions for visualizations.

Augmenting visualization design process using AI. In this work, I aim to address the issues in visualization design by proposing a system that provides design feedback to visualization designer from the eyes of an automated machine. This virtual-human vision-inspired tool aims to serve as a reference tool for visualization designers to improve their designs. This idea is motivated by the concept of *Jack*, a human-motion simulation tool by Norman Badler in 1993 that aims to detect errors about a system or artifact prior to deploying to practice. The goal we are trying to approach is analogous to the intentions of what Jack tries to achieve: to inspect the visualizations from the eyes of the Electronic Eye prior to be presented by the audience. The tool should be capable of looking at the visualization, analyze and detect problems, and eventually provide feedback about visualization artifacts. While the simulation in Badler’s Jack is mostly about motion simulation, our focus is on simulating human perception mechanism using

AI techniques.

1.2 A Virtual, Human Vision-inspired System for Augmenting Visualization Design

This thesis aims to build a virtual human vision-inspired system to augment visualization design. In particular, I focus on three steps to create such a system. First, I devise a method to effectively instill human perception into machines. Second, I find a method to effectively represent the human vision-inspired system. Lastly, based on the analysis of the chart, I create a method to effectively deliver feedback and suggestions to visualization designers.

To that end, this thesis is composed of three works: A SCANNER DEEPLY, PERCEPTUAL PAT, and VISUALIZATIONARY. A SCANNER DEEPLY is a methodology to instill eye movement patterns into a deep learning model. PERCEPTUAL PAT presents a framework to represent human perception in an actionable, pragmatic manner for visualization designers. Finally, VISUALIZATIONARY automates the delivery of perceptual feedback, from image analysis to visualization suggestions. Below, we provide a brief introduction to each work.

1.2.1 Simulation: A Scanner Deeply

Visual perception is a key component of data visualization. Much prior empirical work uses eye movement as a proxy to understand human visual perception. Diverse apparatus and techniques have been proposed to collect eye movements, but there is still no optimal approach. After reviewing 30 prior works, I analyze these works for collecting eye movements based on three axes: (1) the tracker technology used to measure eye movements; (2) the image stimuli

shown to participants; and (3) the collection methodology used to gather the data. Based on this taxonomy, I employ a webcam-based (low-tech requirement) eyetracking approach using task-specific visualizations as the stimuli. The low technology requirement means that virtually anyone can participate, enabling us to collect data at a large scale using crowdsourcing: approximately 12,000 samples in total. Choosing visualization images as stimuli ensures that the eye movements will be specific to perceptual tasks associated with visualization. I use these data to propose A SCANNER DEEPLY, a virtual eyetracker model that, given an image of a visualization, generates a gaze heatmap for that image. I employ a computationally efficient yet powerful convolutional neural network, called *SimpleNet*, for our model. The analysis of our gaze patterns enables us to understand how users grasp the structure of visualized data.

1.2.2 Representation: Perceptual Pat

Designing a visualization is often a process of iterative refinement, where the designer improves a chart over time by adding features, improving encodings, and fixing mistakes. However, effective design requires external critique and evaluation. Unfortunately, such critique is not always available on short notice, and evaluation can be costly. To address this need, I present Perceptual Pat, an extensible suite of AI and computer vision techniques that forms a virtual human visual system for supporting iterative visualization design. The system analyzes snapshots of a visualization using an extensible set of filters—including gaze maps, text recognition, color analysis, etc. Then, it generates a report summarizing the findings. These filters are designed to represent a virtual human vision-inspired system in an actionable and pragmatic manner. The web-based Pat Design Lab provides a version tracking system that enables designers to track

improvements over time. I evaluate its effectiveness by allowing four experienced visualization designers to create visualizations using Pat Design Lab. The results show that Pat Design Lab helps visualization designers by detecting issues in visualizations.

1.2.3 Automation: Visualizationary

Interactive visualization editors empower people to author visualizations without writing code, but they do not guide them in the art and craft of effective visual communication. In this paper, I explore the potential for using an off-the-shelf Large Language Model (LLM) to provide actionable and customized feedback to visualization designers. The implementation, called VISUALIZATIONARY, showcases how ChatGPT can be used in this manner through two components: a preamble of visualization design guidelines and a suite of perceptual filters extracting salient metrics from a visualization image. I present findings from a longitudinal user study involving 13 visualization designers—6 novices, 4 intermediates, and 3 experts—authoring a new visualization from scratch over the course of several days. The results indicate that providing guidance in natural language using an LLM can aid even seasoned designers in refining their visualizations.

1.3 Contributions

Here I list the contributions of each work conducted for the thesis.

A Scanner Deeply A Scanner Deeply explores a methodology to instill human perception into machines. As an example, it tries to create a framework for a virtual eyetracker optimized for use by visualization designers. The first contribution of this work is the study of taxonomy.

We show that collecting eye movement data on chart images using crowdsourcing with low-tech requirements and training the data using deep learning techniques are the desiderata for creating a virtual eyetracker. The second contribution is the framework to create a virtual eyetracker, A Scanner Deeply. In doing so, we identify and address several issues that occurred during the creation of A Scanner Deeply, such as removing noise that hinders the model’s proper training. The third contribution is the patterns learned from the analysis of the salience maps generated by A Scanner Deeply. We observe that the salience patterns are task- and chart-specific. The fourth contribution is the disclosure of the dataset.

Perceptual Pat In Perceptual Pat, we represent a virtual human vision-inspired system to augment the iterative design process of a visualization designer. The first contribution of this work is the representation of a virtual human vision-inspired system using a suite of perceptual filters. We utilize various filters created using computer vision techniques that can analyze charts in an actionable and pragmatic manner. The second contribution is the implementation of Perceptual Pat into a web-based interface called Pat Design Lab. The third contribution is the report on the responses of visualization designers using Perceptual Pat. From four visualization designers, we observe that Perceptual Pat helps improve design by detecting issues in visualizations.

Visualizationary In Visualizationary, we automate the entire process of providing perceptual feedback, from the analysis of an inputted chart to actionable suggestions. The first contribution of this work is the design of the automation process, called ACGT, which comprises analyze, clarify, guide, and track. The second contribution is the implementation of the tool, Visualizationary, using large language models (LLM) and a preamble of design knowledge about visualizations. The third contribution is the evaluation of the tool with individuals possessing varying levels of

visualization expertise. While the tool is designed to assist designers of any expertise level, it is particularly beneficial for those with a certain level of experience in visualization.

1.4 Dissertation Structure

The dissertation is composed in the following manner. In the next section, I provide the related background about the concept of a virtual human visual system. Next, I present Scanner Deeply, a CNN-based gaze prediction model created based on crowdsourced eye movement data. This presents a methodology of infusing human perception for visualizations from crowdsourced human data. It also provides the characteristics of a model trained using crowdsourced eye movement data. Then, I present Perceptual Pat, the virtual human visual system for iterative visualization design. This presents a pragmatic concept of a virtual human visual system. Furthermore, it also shows the reactions of expert visualization designers about the system. Afterwards, I present limitations as well as future directions of the work. Finally, I conclude the work.

1.5 Prior Publications and Authorship

Although I am the principal author of the research in this dissertation, it is essential to acknowledge that the work is the product of collaborative efforts. The key collaborators include Dr. Niklas Elmqvist who is my indispensable advisor as well as my role model and Dr. Sanghyun Hong, who is not only my trusted friend but an insightful mentor. The first paper in this thesis, *the Scanner Deeply*, is done in collaboration with Niklas, Sanghyun, and Sunghyo Chung, my friend as well as colleague from DAVIAN Lab at Korea University. *The Scanner Deeply* [6] is

accepted as a IEEE full paper in IEEE VIS2022' Conference, and is published as a journal paper in IEEE Transactions on Visualization and Computer Graphics (TVCG). The second work, *the Perceptual Pat* [7], is done with Niklas and Sanghyun. *Perceptual Pat* is accepted as a full paper in ACM Conference on Human Factors in Computing Systems (CHI). As of the date of defense, *Visualizationary* is currently under submission to IEEE VIS2024' Conference as a full paper, and is done with Sanghyun and Niklas.

I have chosen to present this dissertation in the first-person plural to emphasize that the work discussed herein would not have been possible without the invaluable contributions of my collaborators.

“Exposure, the effective laying open of the data to display the unanticipated, is to us a major portion of data analysis.”

THE FUTURE OF DATA ANALYSIS,

John Tukey

Chapter 2: Background

This section presents the background about my work. I start by describing methods to facilitate visualization design. Then, I provide works that team with AI for visualization design.

2.1 Facilitating Visualization Design

There have been numerous attempts to facilitate an effective chart design process. First, by creating effective chart authoring tools. Second, by developing recommendation tools. Third, by supporting visualization design as an iterative process.

One approach to facilitating visualization design has been through the creation of effective chart authoring tools. One such tool is Polaris, a visualization tool for authoring large multidimensional datasets [8]. Another category includes web libraries like D3 [9], an expressive tool designed for interactive chart creation in a web browser. Vega-Lite [10] is an expressive grammar based on Wilkinson’s Grammar of Graphics. More recently, Charticulator [11], a PowerBI plug-in, has facilitated the creation of bespoke visual representations.

Another type of tool aims to automate the chart authoring process by directly recommend-

ing charts to users. Voyager [12] enables faceted browsing of recommended charts, chosen based on statistical and perceptual measures. Draco [13] is a recommendation tool where design knowledge is formalized as constraints. Dziban [14] is a Draco-driven recommendation tool that addresses ambiguous recommendations based on the user’s past preferences. There have also been attempts to leverage AI techniques in recommending visualizations, such as VizML [15].

Despite these advancements, not all visualizations can be authored in an automated manner. The visualization design process is intrinsically iterative, and in this context, there have been attempts to aid visualization design throughout this iterative process. McNutt et al. [16] define deceptive, erroneous design practices as *visualization mirages*. Using a technique called metamorphic testing, their work detects these mirages at the visual encoding stage of analysis. Another popular method for assisting the design process uses a linting metaphor to detect errors within visualizations [17–19].

Contribution. Our work is designed to assist the visualization designer’s design process, which is inherently iterative. We aim to provide perceptual feedback from the perspective of a virtual humanvision-inspired system (VHVS), a contribution that distinguishes our approach from existing ones.

2.2 Teaming with AI for Visualization Design

AI is a rapidly evolving field, providing numerous opportunities to use AI techniques in visualization research. Consequently, there is a myriad of research that leverages both machine learning and visualization techniques. Significant interest has been focused on augmenting the explainability [20–26] of machine learning models using visual analytics tools. Another line of

interest is focused on exploiting machine learning techniques for visualizations [27–30]. In the survey paper on Machine Learning for Visualizations (ML4VIS) [31] by Wang et al., the authors highlight machine-human collaboration and the development of user-friendly machine learning for visualizations as some of the ongoing challenges in the field.

Contribution. Our works address the challenges suggested by Wang et al. by exploring issues that arise and finding solutions to provide qualitative feedback for visualization designers, emphasizing human-machine collaboration. It also exemplifies the Human-Centered AI approach [32], where AI techniques are used to augment human capabilities rather than replace them.

Chapter 3: A Scanner Deeply: Predicting Gaze Heatmaps on Visualizations

Using Crowdsourced Eye Movement Data

Your eyes are not just windows to your soul, but also to your ability to read and perceive the visual content on a computer screen. Thus, the ability to detect, track, and predict one's eye movements is used in numerous fields such as computer vision [33, 34], human-computer interaction [35, 36], and natural language processing [37, 38]. This is doubly true for data visualization, where tracking the user's gaze on an interactive chart is key to understanding sophisticated mechanisms behind how humans perceive them [39, 40]. However, current eyetracker hardware is bulky and costly, thus preventing widespread use, and the resulting data is often large in scope and difficult to interpret. Recent progress in machine learning and computer vision has made web-based eyetracking using standard webcams feasible [3, 41, 42]. However, such techniques tend to be less accurate than specialized eyetracking hardware and are sensitive to varying lighting conditions, webcam placement, and even the user's appearance.

At the same time, universally available eyetracking would undoubtedly be highly useful during the design and development phases of a data visualization, allowing the designer to quickly gauge the appearance and salience of a visual representation. What if you could, for example, figure out whether a peak in data stands out, or if a relationship between two items can be seen in the visual clutter? However, to the best of our knowledge, such functionality does not yet exist.

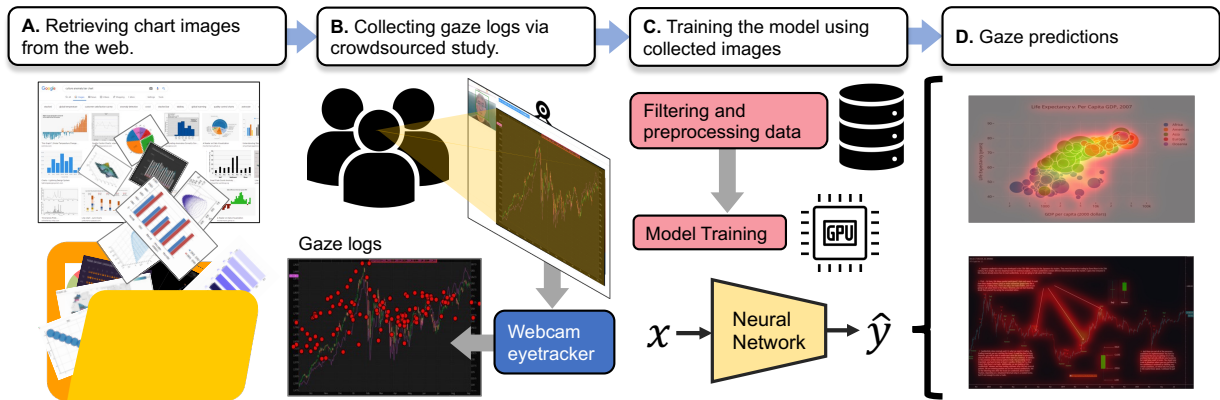


Figure 3.1: **Summary of the Scanner Deeply pipeline.** Pipeline for developing a gaze prediction model that, given an image as an input, produces a saliency map for an image containing a visualization. First, we collect more than 10,000 images that contain one chart per image (A). Second, using the collected images we conduct a crowdsourced study on Amazon Mechanical Turk to gather gaze logs using a webcam eyetracker (B). Third, using the image collection with gaze log annotations, we train a model called SimpleNet, a CNN-based neural network model (C). Gaze predictions are shown in (D) in the form of saliency maps.

We propose a virtual eye tracker using deep learning—a SCANNER DEEPLY—that, given an image of a visualization as an input, automatically generates a gaze heatmap for the image. Fig. 3.1 outlines our approach for developing this deep learning-based eye tracker. We first gathered a large corpus of some 11,000 visualization images from the web. Then we collected eyetracking data at scale by conducting a crowdsourced user study on Amazon Mechanical Turk that leverages an existing webcam-based eyetracking technique [3] to collect gaze logs for our visualization corpus. We used these images and the annotated gaze logs to train a state-of-the-art convolutional neural network-based model [43] that predicts gaze heatmap when an image is given as an input. Based on the images created from Scanner Deeply, we present its qualitative features and evaluate our work using a model trained on the Salicon dataset and the DVS model. We also present a preprocessing method that effectively removes noisy gaze dots that improves the quality of webcam-based gaze dots. Our work demonstrates that gaze patterns on visualizations are task- and domain-specific.

Table 3.1: **Taxonomy of prior work.** We categorize the 30 papers that collect eye movement data with three dimensions: (1) the tracker, (2) image stimulus, and (3) collection methodology. Note that the papers are listed in chronological order. We found that there is no work that employs the same collection approach as ours, i.e., collecting webcam-based crowdsourcing eye movements on chart images.

	Title (shortened)	Year	Eyetracker	Image Stimulus	Collection Methodology
1	A Model of Saliency-Based Visual Attention for Rapid... [44]	1998	eyetracker	natural	lab experiment
2	Using Eye Tracking to Investigate Graph Layout... [45]	2007	eyetracker	charts	lab experiment
3	Visual Perception of Parallel Coordinate Visualizations [46]	2009	eyetracker	charts	lab experiment
4	Learning to Predict Where Humans Look [47]	2009	eyetracker	natural	lab experiment
5	Eye Movements During Mindless Reading... [48]	2010	eyetracker	texts	lab experiment
6	Evaluation of Traditional, Orthogonal, and Radial... [49]	2011	eyetracker	charts	lab experiment
7	Findings while Investigating Visualizations for... [39]	2012	eyetracker	table	lab experiment
8	User-Adaptive Information Visualization - Using Eye... [50]	2013	eyetracker	charts	lab experiment
9	Predicting Affect from Gaze Data During Interaction... [51]	2014	eyetracker	texts	lab experiment
10	A Crowdsourced Alternative to Eye-tracking for... [52]	2015	mouse/cursor	charts	crowdsourcing
11	TurkerGaze: Crowdsourcing Saliency with Webcam... [53]	2015	webcam	natural	crowdsourcing
12	Constructing Models of User and Task Characteristics... [54]	2015	eyetracker	charts	lab experiment
13	SALICON: Saliency in Context [2]	2015	mouse/cursor	natural	lab experiment
14	Predicting Confusion in Information Visualization... [55]	2016	eyetracker	charts	lab experiment
15	Do graph readers prefer the graph type most suited to... [56]	2016	eyetracker	charts	lab experiment
16	WebGazer: Scalable Webcam Eye Tracking Using User... [3]	2016	webcam	natural	crowdsourcing
17	Eye Tracking for Everyone [57]	2016	phone camera	eye gaze	crowdsourcing
18	Beyond Memorability: Visualization Recognition and... [40]	2016	eyetracker	charts	lab experiment
19	Zone out no More: Mitigating Mind Wandering... [58]	2017	eyetracker	texts	lab experiment
20	Learning Visual Importance for Graphic Designs and ... [59]	2017	mouse/cursor	charts	crowdsourcing
21	SearchGazer: Webcam Eye Tracking for Remote... [60]	2017	webcam	web	crowdsourcing
22	Fauxvea: Crowdsourcing Gaze Location Estimates for... [61]	2017	mouse/cursor	charts	crowdsourcing
23	Saliency Revisited: Analysis of Mouse Movements... [62]	2017	mouse/cursor	natural	lab experiment
24	Patterns of Attention: How Visualizations are Read... [63]	2017	eyetracker	charts	lab experiment
25	Revisiting Video Saliency: A Large-scale Benchmark... [64]	2018	eyetrackers	natural	lab experiment
26	Exploring Visual Attention and Saliency Modeling... [65]	2018	eyetrackers	natural	lab experiment
27	Predicting Visual Importance Across Graphic Design... [66]	2020	mouse/cursor	natural/graphics	crowdsourcing
28	Visual Saliency Model Based on Crowdsourcing Eye... [67]	2020	webcam	natural	crowdsourcing
29	TurkEyes: A Web-Based Toolbox for Crowdsourcing... [42]	2020	mouse/cursor/zoom	natural	crowdsourcing
30	Gaze-driven Adaptive Interventions for MSNV... [68]	2021	eyetracker	charts	lab experiment
-	SCANNER DEEPLY (OUR WORK)	2022	webcam	charts	crowdsourcing

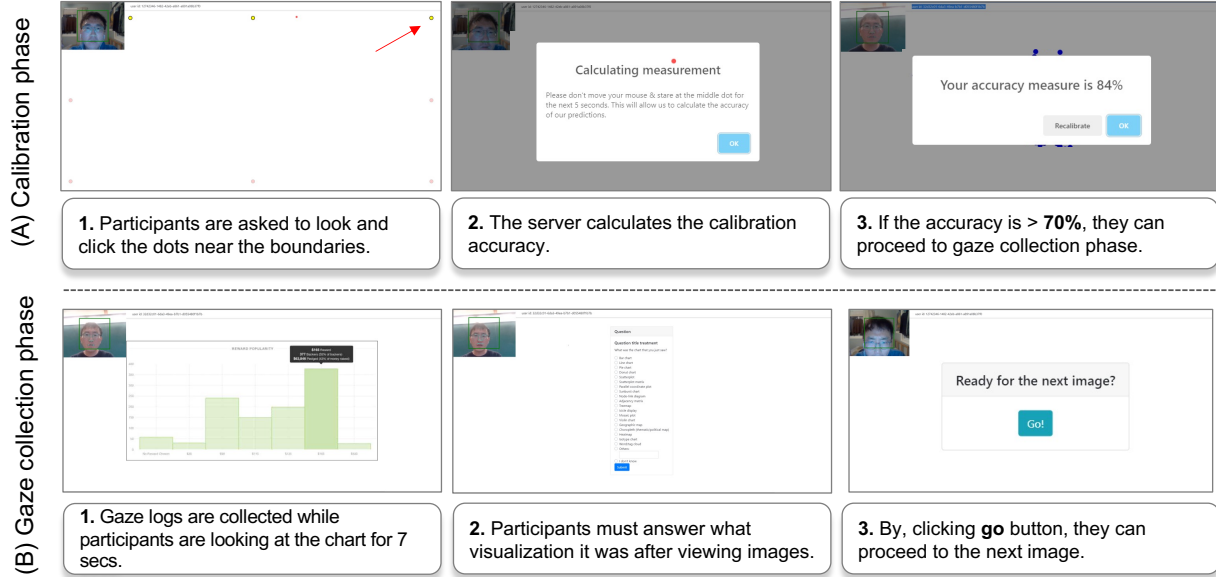


Figure 3.2: **Calibration and gaze collection.** Participants click on 8 dots that are placed near the boundaries of the screen. Each dot turns yellow once the user has clicked on it five times (A-1). Then, the calibration accuracy is measured by making users look at the middle dot at the center of the screen (A-2). If the accuracy is above 70%, then participants can proceed to the gaze collection phase (A-3). In the gaze collection phase, participants are asked to look at the chart for 7 seconds, during which time the user’s gaze is tracked (B-1). Then, they are asked to answer what type of chart the image is (B-2). After they submit the answer, participants look at the next image by clicking “Go” button at the question “Ready for the next image?” We customize the code WebGazer developed by Papoutsaki et al. [3] in creating the website.

3.1 Overview

We review works that study visual perception via human eye movements. As summarized in Table 3.1, diverse apparatus and techniques have been proposed to collect eye movements data. We taxonomize those works based on three axes: tracker technology, image stimulus, and collection methodology. We detail why some choices are not practically desirable given the recent research trends. We conclude this section with a discussion on our choices.

3.1.1 Understanding Visual Perception via Eye Movements

Research on understanding human perception using eye movement started more than three decades ago [44], and the methods for collecting eye movements have made advances over time. One form of early works that involve studies with eye movement data explore different human reactions and patterns as a lab study. Eye movement data are typically collected using eyetracking hardware. Examples of these experiments include understanding mind wandering patterns when reading text [48], how people react under tabular visualizations [39], and how people make sense of unfamiliar visualizations [69]. Another form of early work aims at developing gaze prediction models. First introduced by Itti et al. [44], there has been many attempts to build models predicting gaze views of natural images, e.g., models using concepts from information theory to predict salience on natural images [70, 71].

As eyetracker started to gain more popularity in human perception research than the past, there has been growing demand for gaze prediction models with high quality. This brought light to numerous data-driven gaze prediction models [72, 73] and also to large-scale gaze logs on natural image datasets [47]. Two most commonly used image datasets for training and evaluating saliency models are (1) CAT2000 benchmark [74] and (2) Salicon [2]. CAT2000 contains 2,000 images from 20 different categories, and Salicon contains 10,000 images drawn from MS COCO [75]. Salicon provides highly varied and natural images along with ground-truth fixation annotations. With rapidly developing deep learning techniques, the performance of saliency maps on the benchmark datasets have also rapidly ameliorated [76–78]. Furthermore, as crowd-sourced platforms became available, various methods have been developed to democratize the eyetracking process, such as cursor-based eyetrackers [41], or webcam-based eyetrackers [3, 53].

Initial work in the visualization community on gaze prediction builds models using natural image datasets [79]. Inspired by the works showing that humans focus on texts while analyzing charts, Matzen et al. proposed DVS [1], a gaze prediction model that utilizes a linear combination of a model based on natural images [70] and text optimizers [1]. However, it has been unclear whether those models based on natural images are effective on visualization images. In evaluation (§3.4.1), we show that such a model is less effective than our models trained on a visualization dataset, which necessitates developing a task-specific—i.e., focusing on data visualization—data corpus. It also has been unknown whether neural networks that improve the performance of gaze prediction tasks in natural image domains, are effective for visualization images, especially for charts. Our work aims at addressing this knowledge gap by collecting a large-scale gaze dataset of charts and training neural networks for gaze prediction on the data.

3.1.2 Choice of Apparatus and Techniques in Prior Work

Building gaze prediction models requires a corpus of eye movement data. Prior work therefore utilizes diverse apparatus and techniques for collecting datasets. Here, we review 30 prior works and assess the advantages and disadvantages of their collection methods. We collect those papers from visualization (IEEE TVCG, IEEE VIS, etc.), computer vision (IEEE CVPR, IEEE ICCV, etc.), and human-computer interaction (ACM CHI, ACM UIST, etc.). We consider studies comprehensively, i.e., they deal with eye movements on visualizations, natural images, texts, and webpages. We then evaluate their choices based on three axes: (1) the *tracker* that is used to track eye movement of a participant, (2) the *image stimulus*, or the type of images that is shown to participants, and (3) the *collection methodology* that is used to gather the data from participants. We

have listed those works in Table [3.1](#).

Eyetracker. Understanding one’s eye movement patterns provide us with various information about the attention the person is focusing on. For example, the fixation data about an image play as indicators of important regions within the image. In the field of visualization, by knowing which part of the chart would people’s attention be most focused, the visualization designer can slightly change her design to better meet her intentions. There exist three types of methods that are used for eyetracking: specialized eyetracking hardware, indirect measurement methods, and general-purpose webcams.

To begin with, the benefit of eyetracking hardware is that it provides an accurate measurement of one’s eye gaze. As a result, it has been extensively used in lab experiments. However, the quality of gaze dots are largely dependent on their prices. The performance of some of the inexpensive ones is somewhat questionable, and high-end eyetrackers can cost upwards of \$20,000. For this reason, and also because of the difficulty in recruiting participants for laboratory sessions, experiments involving eyetracking devices are generally known to be expensive.

As an alternative, there have been attempts to collect gaze information without directly tracking the user’s eyes. One approach is the cursor-based gaze collection technique. This is based on the assumption that the movement of cursor is correlated with the eye movement when looking at a screen [\[80\]](#)—or at least that participants can reliably be asked to use the mouse cursor in this way. Cursor-based eyetrackers are effective in conveying the person’s location of visual attention [\[80\]](#). Because of this characteristics, various research collects gaze dots using this approach [\[2, 41, 52\]](#). However, cursor-based approaches do have limits and are not a completely accurate replacement for eyetracking devices. It loses some of its information, because it cannot

track the person’s eye when she is moving her eyes without moving the cursor.

There is another alternative that is being studied—the use of general-purpose webcams as an eyetracking method. However, webcams have one infamous hurdle that needs to be addressed: consistent calibration of the eye to the screen. This is mainly because calibration in webcams are very sensitive to even small head motions. Jiang et al. [2] argue that collecting large-scale dataset via general-purpose webcams is not possible, especially in an uncontrolled setting. Because of this, with current technology, gaze collection can only be maintained for a short period of time after calibration [53]. Furthermore, although it is still imperfect technology, we think that it is by far the best method to measure directly one’s eyes, affordably, and still yield large-size data.

Image Stimulus. Two types of data can be used to predict gaze for visualization studies: visualization images, and natural images.

For the former, stimulus is often confined to a particular type of chart (e.g., node-link diagrams, parallel coordinate diagrams, etc.), and the size of the data corpus is typically small, not exceeding 300 images. Recently, a large-scale visualization dataset, called the MASSVIS dataset [81], has been developed. The dataset contains more than 5,000 different types of static visualizations from four topics: government, infographic, news, and science.

Natural image datasets are also referenced to imitate gaze dots about chart images. For example, despite being collected on a different domain of images and not intended to be used for visualization images, the reaction to low-level features (e.g., color, contrast, motion, etc.) of an image in natural image datasets and chart images is similar on certain tasks (e.g., exploration tasks) [65]. Furthermore, since early large-scale gaze datasets are collected using natural image datasets as the stimulus, their results have often been deployed in predicting saliency in

visualizations.

Collection Methodology. There are two main choices for collection methodology: lab experiment and crowdsourcing.

Lab experiments are optimal for collecting small-scale but high-quality eye movement data. As mentioned before, these lab experiments are mainly done using eyetracking hardware. The accuracy and quality of eyetracking dots are high as it deploys a specialized eyetracker and the experiment can be fully controlled. However, collecting gaze dots via lab experiments is expensive, impractical and time-consuming to involve a large number of participants.

Crowdsourced platforms such as Amazon Mechanical Turk (AMT) or Innocentive are beneficial in that it is easier to recruit at a relatively affordable price. Consequently, many researchers use crowdsourced platforms to obtain human-involved datasets in a large scale. However, existing technologies for tracking eye movements are not as accurate as real eyetracking hardware.

3.1.3 Our Approach

Based on our assessments, the most desirable approach is to develop a gaze prediction model that provides the highest accuracy, but at the same time using the least possible resources (i.e., reducing time and cost). Neural network models trained on large-scale natural image datasets achieves superior performance over other methods. We therefore aim to apply this approach to visualization in order to construct a gaze prediction model that we call a SCANNER DEEPLY.

To do this, we choose to gather *chart images* as our *image stimulus*, as we hypothesize this will lead to a more specialized model with better performance for chart input. For the *collection*

methodology, we choose to use a *crowdsourced platform* to be able to collect sufficiently large number of data samples for training neural network models. To reduce the confusion of a model, we only collect eyetracking data for images that contain a single static chart.

We set a task that can provide answers on the generability of tasks. Matzen et al. [1], while describing the DVS, mention the possibility of a general-purpose gaze prediction model. However, several researchers provide empirical evidence that gaze patterns are task-specific. For example, Yarbus’ [82] and Michal and Franconeri’s [83] works on gaze research suggest that human attentions are guided by the task she is conducting. Prior work [1] showed that the ideal visualization has a strong overlap between the regions (1) that are most likely to draw the viewer’s attention (bottom-up) and (2) the regions that convey important information (top-down) about a task. We choose a task that satisfies both conditions and can also be evaluated with a gaze dataset collected in a short time (7 seconds). Specifically, we ask participants to figure what type of chart it is after they view the chart for 7 seconds.

Another question remains—it is not clear how large the image stimulus dataset should be to yield a sufficient amount of eye movement data. To the best of our knowledge, the largest known visualization dataset is the MASSVIS dataset [81] which contains 5,000 static visualization images. However, the dataset is not suitable for our task as images in the infographics category have more than two charts per image, and the subset includes images far smaller than 5,000. For comparison, the Salicon dataset [2] has gaze annotations for 10,000 images, while they are natural images. It means the amount of images in MASSVIS is not sufficient for training neural networks. To that end, we decide to create our own image dataset that contains 10,000 images.

For the *tracker*, we choose a low-technology requirement consistent with our crowdsourced platform. Between cursor-based method and webcams, we choose to go with webcams. The

reason is two-fold. First, webcams are the only method to collect gaze dots that are actually directly measured from one’s eyes in a crowdsourced platform. The argument in this decision is not at all in manifesting the superiority of current webcam-based gaze collection over other methods. Our intention is just to reinforce that it is a choice worth studying because of the benefits we would gain if conducted successfully. Second, the use of webcams as the tracker, chart images as the stimulus on a crowdsourced platform is a choice that has not yet been investigated and verified in the literature. This may be because of the intrinsic issue that there may be noise in gaze dots collected from webcams and that consequently the result is unpredictable. However, if we can devise a method that can successfully ease the impact of noise and successfully train the model, then this can also lead to another contribution.

Given the gaze dataset obtained from crowdsourcing, we train SimpleNet, a neural network devised by Reddy et al. [43]. SimpleNet is a state-of-the-art saliency map generating model that utilizes convolutional neural network architectures (e.g., ResNets [84]), to generate gaze heatmap. It requires less computational resources to train or predict than existing networks designed for gaze prediction.

3.2 Task-Specific Crowdsourced Data Collection

Here, we delineate the steps taken to collect task-specific gaze dots using webcams on a crowdsourced platform. We first describe how we crawled chart images from the web. Then we describe the processes taken to conduct the crowdsourced study on Mechanical Turk.

Table 3.2: **Example queries for collecting chart images from the web.** Each query (on the right) is a combination of two keywords from our topical analysis of papers (on the left) and a chart type (in the middle).

Topic Keywords	Chart	Search Query
culture, anomaly	bar	culture anomaly bar chart
normal, politics	line	politics normal line chart
artist, history	pie	history artist pie chart
global, media	tree	global media tree diagram
meeting, crime	heatmap	meeting crime heatmap
sports, outlier	spider	sports outlier spider map
newyork, time	bubble	newyork time bubble chart
market, medical	scatter	market medical scatterplot
museum, weather	violin	museum weather violin

3.2.1 Visualization Image Collection

The goal in collecting chart images is to collect a dataset that aligns with the distribution of charts that can be found in the web, so that the trained model becomes as versatile as possible. To collect crowdsourced eyetracking data at scale, we first needed a large-scale dataset of visualization images. While the large-scale MASSVIS dataset contains 5,000 images, this may not be sufficient for training a neural network. Furthermore, the dataset also contains images containing multiple individual charts. We base our scale on the Salicon dataset, which contains 10,000 gaze dots. Accordingly, we choose 10,000 images as our target quantity.

We gather images by querying Google Image¹ using three keywords each time, where two keywords are chosen from 379 keywords derived from the topical analysis [85] of papers in scientific communities over the past 10 years (2011–2020), and the last keyword stands for the type of charts (*e.g.*, line chart, bar chart, or heatmap). Examples of these keywords are shown in Table 3.2. Through a series of search queries, we gathered 280,000 images from the web. These

¹<https://images.google.com>

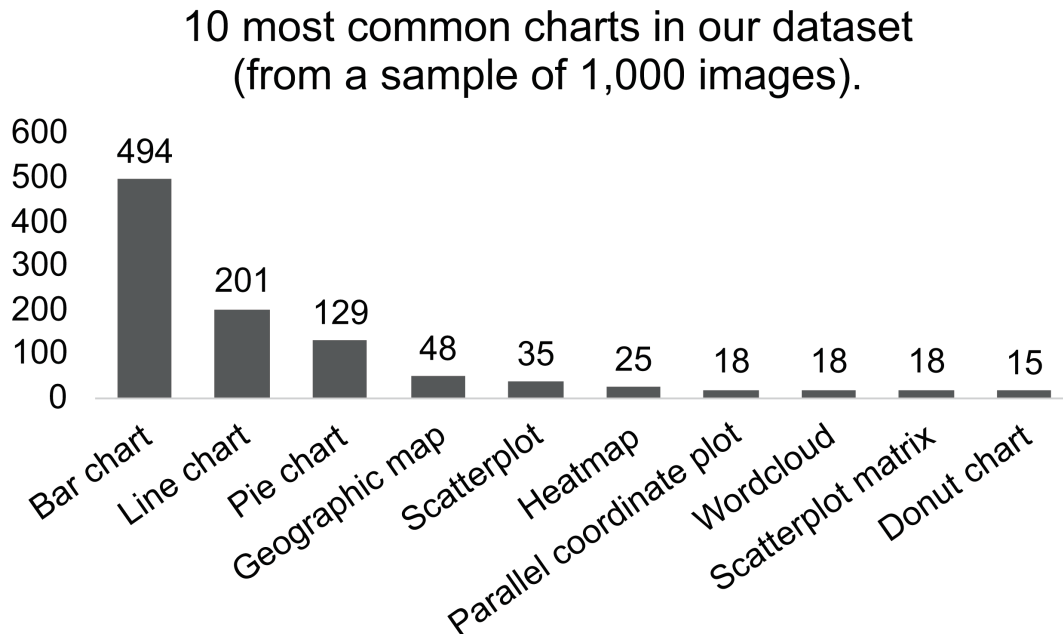


Figure 3.3: **Chart type popularity.** We show the 10 most popular charts from a set of 1,000 samples randomly chosen from our image collection. This provides an estimate on the chart type distribution in our dataset.

images contain not only charts, but also natural images pertaining to the topic. We only kept the images that (1) have one chart in an image, (2) both width and height of an image are larger than 400 pixels, (3) have heights or widths less than 4 times of the other, and (4) are without texts whose sizes are unreadably small. We also removed all duplicates.

In the end, we were able to retrieve 10,960 chart images. To roughly identify the distribution of charts, we randomly sampled 1,000 of the images, and counted them by their chart types. Fig. 3.3 shows the 10 most frequent chart types in a random sample of 1,000 images in our collection. We can observe that bar charts, line charts, and pie charts are the three most commonly used graphs on the web.

3.2.2 Gaze Collection Setup

We collect eyetracking data using crowdsourcing on AMT. We design the procedure with three goals in mind: (1) obtaining high quality gaze dots, (2) obtaining large dataset, and (3) lessening the burden of participants. The task involved participants identifying the kind of chart after looking at the image for a few seconds; while they did so, we tracked their gaze using a webcam eyetracker.

Eyetracking Mechanism We created our experimental platform with the help of WebGazer, an eyetracking technique developed by Papoutsaki et al. [3]. Prior to finalizing our experiment environment, we conducted pilot studies with members of our research group at the University of Maryland to find the optimal settings. We identified two issues in deploying general-purpose webcams as eyetrackers: (1) the sensitivity and accuracy for webcam eyetrackers, and (2) maintaining a high level of concentration from participants.

Calibration in webcams is sensitive to slight movements of the head. Even state-of-the-art webcam eyetrackers exhibit poor robustness against changing head posture. This is exacerbated when testing in the wild, i.e., using a crowdsourcing platform on the internet. During their experiment with webcams, Xu et al. [53] note that eyetracking requires frequent calibration, and to minimize the number, the viewing time must be short. To alleviate this problem, we limit our eyetracked tasks to 7 seconds per image and conduct calibration every 6 images.

Secondly, to check if participants are concentrating, we add attention trials throughout the experiment to determine if participants are concentrating. We select 400 attention images. These attention images are all bar charts given the assumption that all participants know what a bar

chart is, and hence can answer questions without difficulty. Furthermore, as Fig. 3.3 shows, the bar chart is also the most common visual representation in our dataset. If the user does not correctly answer the attention trial, then we stop the participant from further proceeding with the experiment. This fact—that participants remain focused on the trials—is clearly communicated at the outset of the experiment.

Participants. We recruit participants who are fluent in English, do not have color blindness or any other kind of color vision deficiency, and are at least 18 years old. We let participants be aware that the experiment is about charts. Furthermore, we ask them to use computing device with a webcam, have monitors with a screen resolution or 1280×720 or higher, and use Google Chrome or Microsoft Edge web browsers (these requirements are imposed by our eyetracking software).

Apparatus. We build a website for collecting data while participants perform pre-defined tasks. While they are doing so, we track their eye movements. We use JavaScript and jQuery to build our website and use Python Flask v1.1.1 to run it on our university-hosted virtual machine (VM). We allocate 2 GB RAM and 32 GB storage to the VM.

3.2.3 Participant Task

Below we describe steps taken for collecting gaze dots from participants on a crowdsourced platform.

Getting Consent. Participants are first presented with a consent form, including a brief introduction to what the study is about and how the study is run. Afterwards, we lead participants to a

website hosted on our server. The server was created by the authors and is run on a Linux-based virtualization environment provided by the department of Computer Science at the University of Maryland.

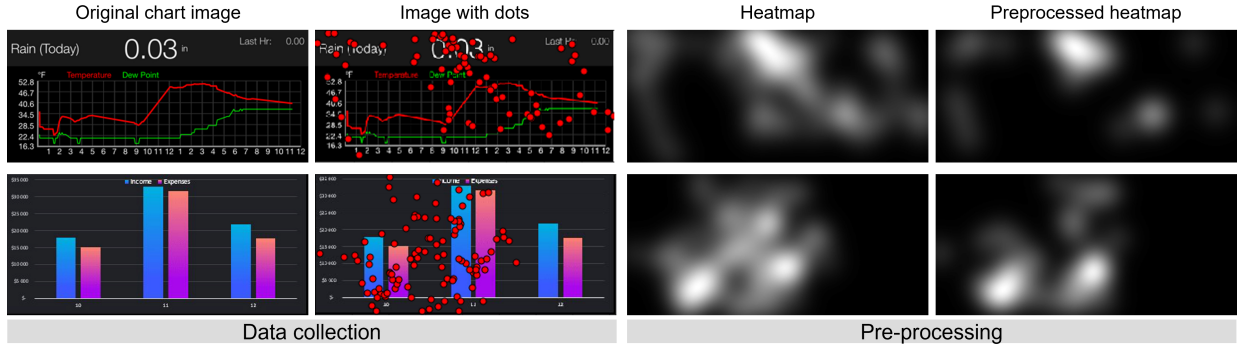


Figure 3.4: **Illustration of procedures for constructing oracle heatmaps.** We use the final heatmaps for training our neural networks. Gaze dots collected by using webcams contain rich information that enables further analysis, e.g., temporal changes in human visual perception (see §3.4.4 for the analyses), but they are also noisy to learn. We address this challenge by carefully pre-processing gaze dots. The second column shows the dots in the raw data from webcams, and we blur dots and remove some of them not in the human’s area of focus (see §3.3.2 for details).

Calibration. Prior to conducting the experiment we provide several suggestions to participants to help them get past the accuracy threshold during the calibration phase: (1) conducting the experiment in a well-lit environment, (2) situating their heads within the green box shown on the camera view, and (3) trying not to change their head posture during the experiment, as it will decrease the calibration accuracy.

After participants allow access to the camera, the calibration can proceed. Fig. 3.2 (A) illustrates the steps required for calibration. Calibration takes place in the following manner: Participants are asked to hold their head steady and place their head on the box shown in the camera view. Then, they are asked to synchronously look at and click at 8 dots placed near the margins of the browser five times each. The gaze dot, or the estimated gaze location of

the participant, is represented by a small moving red-colored dot. Every time a target button is clicked, the opacity of it gets lower until it eventually turns into yellow to signal that it has been fully clicked 5 times. These clicks adjust the calibration between the participant's eyes and the browser window. After all dots turn yellow, a new red-colored button shows up at the center of the browser to measure the accuracy of the calibration. The participants are required to look at the dot until it turns yellow. Accuracy is measured by the proportion of gaze dots that are placed near the center and those that are not. As described in the previous part, if that proportion is higher than 70%, then participants can proceed to the gaze collection phase. If the accuracy is below 70%, then participants must repeat the process until the accuracy rises above the threshold.

Data Collection. The next step is where the gaze log collection process starts. Fig. 3.2 (B) explains the steps taken to collect gaze logs. Once participants click the button “Go” from the question “Are you ready?,” an image appears at the center of the screen. The image appears for 7 seconds. During the 7-second period, gaze logs are collected at a refresh rate of 20 Hz. After 7 seconds have passed, the screen changes into a 20-multiple choice question that polls the chart type. The participants can choose to answer between the choices, select “I don’t know,” or provide a new answer after choosing “other.”

As stated before, we compose one unit block as one calibration followed by 6 gaze trials. The intention is to keep one human intelligence task (HIT) under 3 minutes, including the calibration. From our pilot study, collecting gaze logs from a full block averaged around 1 minute and 10 seconds. Among the 6 trials in a block, one is an attention trial and 5 are used for the experiment. After each HIT is complete, we collect a packet from the server that contains information on (1) the gaze log information, (2) width, height, and size of the browser (e.g., the exact

displacement of the image, the size of the screen, etc.), (3) answers to the questions asked, and (4) the name of the image. We do not capture nor reference videos from the webcam; the latter is clearly stated to participants prior to participating in the experiment.

One HIT is composed of three blocks. Based on our pilot study, it took approximately 3 minutes and 33 seconds to complete one HIT. Based on the targeted rate of \$15 per hour upon successful completion of the experiment, we pay each participant \$0.90 per HIT. For those that perform more than one HIT, we compensate them in the form of bonuses, at the rate of \$0.30 per block. We gave compensation to any participants that conducted the experiment even if only partially. We paid participants no more than five days after the date of submission.

3.2.4 Collection Results

For the 10,960 images, we ran this collection process until we had at least one annotation per image, resulting in 12,504 gaze dots. It took 60 days to collect the data. We exclude the gaze dots from participants who submitted incorrect identification numbers. The dataset consists of gaze logs from 9,157 correct responses and 3,347 incorrect responses. Examples of gaze logs collected from our study are shown in Fig. 3.4. The success rate of each HIT task was 71.2%. The unsuccessful HITs are due to (1) incompleting tasks (21.3%) and (2) false answers (7.5%).

3.3 A Scanner Deeply

We now propose a SCANNER DEEPLY, a virtual eyetracker that utilizes a deep neural network (DNN) for gaze prediction. Scanner Deeply takes a visualization image as an input and automatically generates a gaze heatmap for the image. We first discuss our choice of a

DNN: we employ SimpleNet [43], a compact DNN architecture designed for gaze prediction. We then describe how we preprocess our dataset to prepare the training data, i.e., data filtration and saliency map generation. We finally describe how we train SimpleNet on the preprocessed data.

3.3.1 SimpleNet: A DNN for Gaze Prediction

We have two criteria for choosing a DNN architecture. First, the architecture should be fast and computationally efficient at inference time. Most DNN architectures that are used in prior work [77, 86] for gaze prediction are complex, i.e., they contain millions of model parameters, which increases the operational costs. To run inferences with those models, we require special hardware (e.g., GPUs or hardware accelerators). Reducing a DNN’s post-training operations thus allows users and practitioners to deploy the Scanner Deeply to diverse computing environments, ranging from servers and personal computers to devices with limited computational resources, e.g., IoT or mobile devices. Second, while reducing the costs of post-training operations, we choose the architecture that provides state-of-the-art performance in gaze prediction tasks. Deep and complex architectures [87] typically offer better performance. However, we aim to find a shallow, simpler network that can achieve on-par performance. Considering the criteria, we employ SimpleNet to implement the Scanner Deeply.

Fig. 3.5 illustrates the SimpleNet architecture adapted for our pipeline. SimpleNet employs an encoder-decoder architecture. The encoder extracts latent representations (often referred to as features) from an input, and the decoder reconstructs the input from the latent vectors. In the figure, SimpleNet extracts a 2048-dimensional vector from a visualization image as a latent

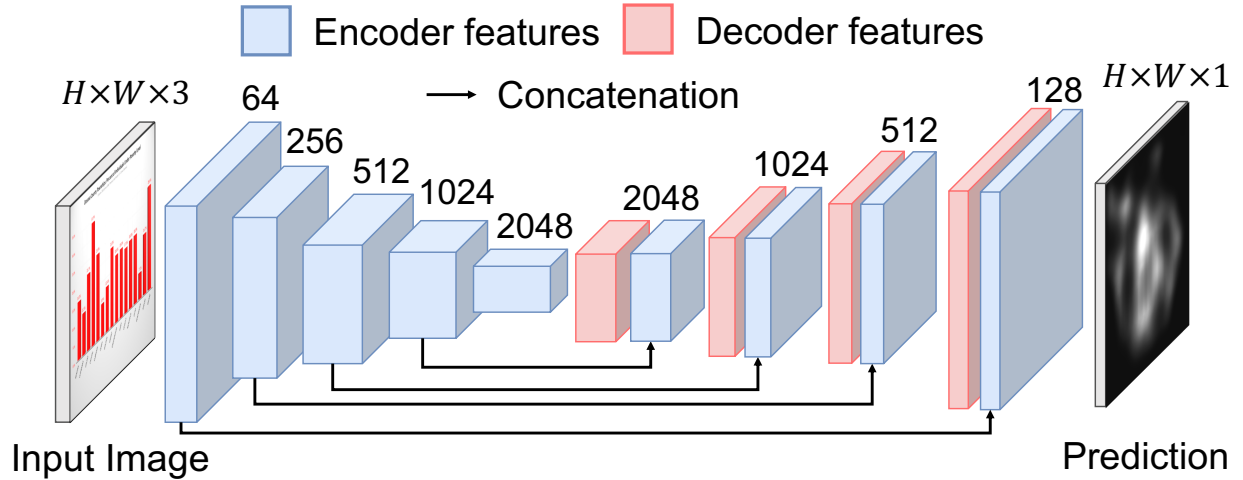


Figure 3.5: **Our SimpleNet architecture.** The pipeline accepts image input and produces an output image. In our case, the input is a visualization and the output is the gaze map. SimpleNet follows an encoder-decoder structure similar to U-Net [4]. Compared to DNNs from prior work [2], SimpleNet requires less computation.

representation. From this vector, the decoder architecture generates a gaze map. Multiple image classification models, such as VGGNet [88], ResNet [84], and PNASNet [89], can be used as an encoder; we choose ResNet because it offers the highest performance. The decoder is composed of two deconvolutional layers. SimpleNet utilizes the U-Net structure [4], which helps to improve the performance further by incorporating the information from earlier layers when the decoder reconstructs a gaze map.

3.3.2 Preprocessing Webcam-based Eyetracking Data

Even with our efforts to maintain a high level of calibration during gaze collection process, the resulting gaze dots from webcam-based eyetrackers contain noise. Noise is mainly attributed to the dispersion of gaze dots. Such dispersion happens as participants make diminutive movements of their heads, or because poor or shifting light conditions introduce errors in the eyetracking software. At worst, noisy ones exist in locations far from where the chart is located

within an image, and they prevent the DNNs from learning patterns in the data. (see Table 3.3)

To address this issue, we filter out gaze dots not in a *saliency zone*. Following the prior work [70, 90, 91], we define the saliency zone as areas in an image that are highly disparate in RGB values in contrast to its neighboring pixels. We preprocess gaze dots as follows: At each pixel location, we first compute color differences between the location and all the neighboring pixels within a 5-pixel radius. We perform this step over all the pixels in an image and remove gaze dots at each pixel location where the sum of the color differences is less than 10 (10 is the threshold we empirically find). We then blur the fixations by using a Gaussian filter that has a standard deviation of 5 and set the boundaries of the salient zone. In Fig. 3.4, we show examples of our preprocessed gaze dots. We define noise as the eye movement data that do not belong in the saliency zone and use only those that exist in the saliency zone. There is an implied prior behind this mechanism: this process assumes that salience tends to focus near zones where there is a change in salience. Using this method, the model automatically learns from the training dataset the zones where there is a change in pixels. Additionally, the model learns from the data the salient parts of these zones in each chart image. We exclude 160 images that this filtration process removes all the gaze dots from our dataset, leading to 12,344 gaze maps, consisting of 9,040 correct and 3,304 are incorrect ones. Note that our preprocessing does not change the input resolution.

3.3.3 Training SimpleNet on Our Crowdsourced Dataset

We implement Scanner Deeply with Python 3.9 and PyTorch v1.10.² To train our SimpleNet models, we use a machine equipped with Intel i7-6700K CPU with 32GB RAM and 2

²<https://pytorch.org>

NVIDIA GTX 1080 Ti GPUs. Note that, once trained, we only use CPUs to offer gaze prediction.

Datasets. We compose our dataset as follows: Among the 9,040 correctly-responded visualization images, we randomly pick 70% of them as our training data and use the rest 30% as the testing set. This split leads to 6,328 training and 2,712 testing images. Next, we pair each visualization image with a gaze map preprocessed by the method described in §3.3.2. Those maps are the oracles the model should generate. All the visualization images and saliency maps are scaled to 256×256 . Our model produces gaze predictions with the same size, and we re-scale them back to the original resolutions for visualization.

Objective Function. We train SimpleNet to minimize the perceptual difference between gaze predictions and the oracle saliency maps. To measure the difference, we employ the Kullback-Leibler (KL) divergence, an objective function commonly-used in literature [92]. We also examine other metrics for measuring perceptual differences, e.g., ℓ_p -distances or normalized scanpath saliency (NSS) [93], proposed by prior work [94]. While our model minimizes those metrics, we observe in our manual analysis that gaze predictions generated with KL divergence are better than the cases of using others.

Hyper-parameters. We use the Adam [95] optimizer to train our models. We set the batch size to 32 and the learning rate to 10^{-4} . We also set the weight decay to 10^{-4} . We train our models for 40 epochs; at each epoch, we compute the KL divergence of our model on the testing set and store the one that minimizes the metric over the 40.

3.4 Evaluation

Here, we evaluate the Scanner Deeply. We first show the prediction performance of the Scanner Deeply and the baselines (§3.4.1). We then compare the gaze predictions qualitatively to illustrate unique characteristics the Scanner Deeply captures (§3.4.2 and §3.4.3). We lastly show the benefit of using webcam-based eyetackers by analyzing temporal changes in human visual perception with the Scanner Deeply (§3.4.4).

3.4.1 Prediction Performance of the Scanner Deeply

We evaluate the performance of the Scanner Deeply. The purpose of this experiment is to show that for predicting gaze heatmaps for a specific task, it is desirable to train models on a task-specific dataset. To this end, we train (1) SimpleNet on Salicon [2], gaze maps for a set of natural images and (2) use the DVS model [1]³ designed to perform generally well on predicting gaze maps for visualizations. We employ five metrics for measuring perceptual similarity between gaze predictions and gaze maps on our 2,712 testing images: two location-based metrics, i.e., AUC-Judd (AUC-J) [47] and NSS, and three similarity-based metrics, i.e., KL-Div, SIM, and CC [96]. Except for KL-Div, the higher a metric is, the more gaze predictions are similar to oracles. Table 3.3 shows our results.

Scanner Deeply shows better performance over the two baselines. We first observe that the Scanner Deeply exhibits 5–15% higher performance than SimpleNet trained on the Salicon dataset. Considering that the training dataset is the only difference between the two models, it is important to use the dataset collected from the same domain (i.e., visualization) for high-quality

³<https://github.com/LauraMatzen/DVS>

Table 3.3: **Performance evaluation of Scanner Deeply by comparing it with gaze prediction models proposed by prior work.** We show the performance using five metrics. We report each metric’s mean and standard deviation over the five runs. The value shown on top is the mean, and on the bracket below is the standard deviation over the five runs. We compare with DVS [1] and Salicon [2]. The only difference between Salicon and the Scanner Deeply is in the dataset. We show that a model using datasets collected for the specific task and domain on a specific task can show improved performance over a general-purpose gaze prediction model. We also compare ours with Scanner Deeply before pre-processing. This is represented as Ours[†].

Metric		Ours	Ours [†]	DVS	Salicon
Location-based	(↑) AUC-J	0.784	0.751	0.737	0.741
		(0.002)	(0.002)	(0.005)	(0.002)
	(↑) NSS	0.716	0.625	0.489	0.537
		(0.004)	(0.006)	(0.005)	(0.009)
Distribution-based	(↓) KL-Div.	1.283	1.333	1.507	1.443
		(0.035)	(0.053)	(0.022)	(0.063)
	(↑) CC	0.403	0.373	0.276	0.301
		(0.166)	(0.186)	(0.136)	(0.193)
	(↑) SIM	0.429	0.388	0.373	0.392
		(0.002)	(0.014)	(0.012)	(0.003)

[†] SimpleNet trained on the dataset w/o the preprocessing step in §3.3.2.

gaze prediction. Compared with the DVS model, we find that the Scanner Deeply achieves 8–18% improvements. This result implies that within the same domain, a model trained on a task-dependent dataset can perform better on a specific task than a model built for a general-purpose prediction.

We also assess the impact of the preprocessing step on the performance of the Scanner Deeply (Ours vs. Ours[†]). Across the board, the similarity metrics from the models trained on preprocessed gaze maps are better than those trained on un-preprocessed maps. This confirms our hypothesis that filtering out gaze dots in the image area with no stimulus helps a model focus more on the important low-level features. We find that the improvements are larger for the location-based similarity metrics (i.e., AUC-J and NSS), which implies that preprocessing encourages a model to ignore the unimportant parts in visualization.

3.4.2 How Does the Scanner Deeply Perceive Charts?

We now present unique characteristics of how the Scanner Deeply perceives visualization images compared with our baselines. We make this comparison by analyzing how those three models perceive three different stimuli within an image: (1) charts (visualized data information), (2) textual information, and (3) low-level features [65]. In Fig. 3.6, we illustrate gaze predictions generated by three models for four images.

Chart Perception. A predominant characteristic of gaze predictions produced by the Scanner Deeply is that it focuses on the shape (or structures) of charts within images, while the baseline models focus more on the texts. Fig. 3.6 shows that gaze predictions from DVS and SimpleNet models, trained with Salicon, highlight legends, numbers, or titles. In contrast, the Scanner Deeply focuses on a slice in the pie chart (A), peaks in the line chart (B), branching points in the sankey tree (C), and edges in the geographical maps (D). This means that the Scanner Deeply has the ability to emphasize different parts of images depending on the visualization tasks. The task here is to determine the type of a chart. It also means that in our data collection, participants have looked at the entire chart to perform the task, not focusing only on a specific part (see in §3.4.3 and 3.4.4 for our detailed analysis).

If we take a closer look at the gaze predictions from the DVS model (see the figures in the middle column), the model explicitly captures texts the most. The result implies that DVS is designed to capture both the low-level features (via the GBVS model) and texts (via text optimizers), but in fact, the text optimizers seem to be a dominant component. Surprisingly, from the figures in (B), DVS predicts that humans will mainly focus on reading the x-/y-axis. However, it

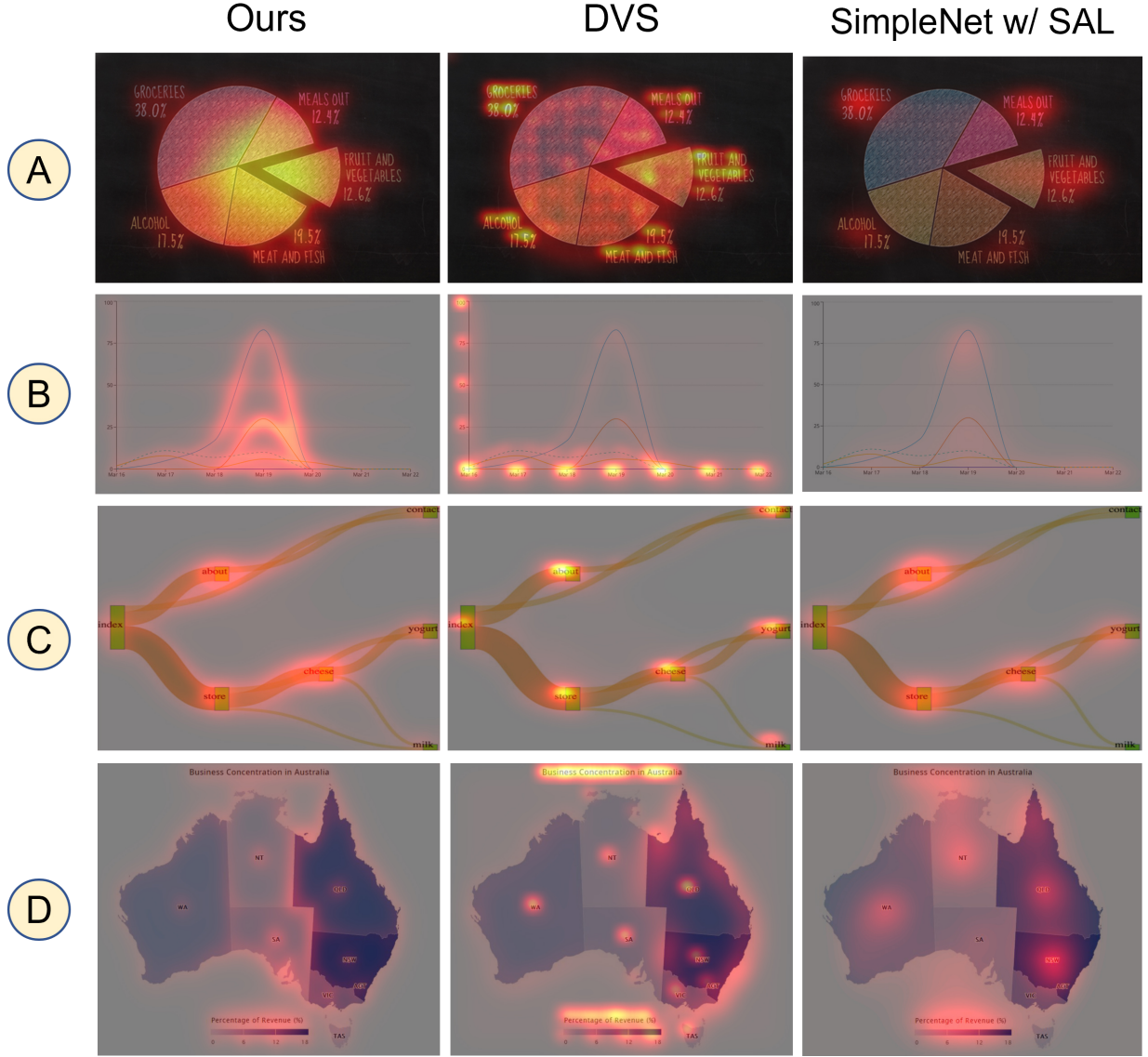


Figure 3.6: **Visualization of gaze predictions.** We show the gaze predictions of three charts from ScannerDeeply, DVS, and SimpleNet trained on the Salicon dataset. DVS and Salicon focus more on text, while our model focuses more on distinct areas in a chart.

is unlikely to be true as we use charts for reducing the perceptual complexity.

We further show that the model learned from natural image datasets (e.g., Salicon) focuses on low-level features the most. If the shape of the chart is easily noticeable with respect to its environment, the model focuses on the chart (see the figures in rows (A) and (C)); otherwise, it emphasizes other areas, e.g., texts. Here, being noticeable means how much contrast the chart

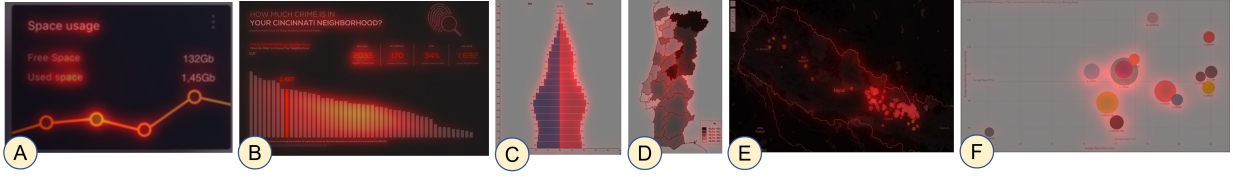


Figure 3.7: **Gaze predictions from ScannerDeeply for different types of visualizations.** We show examples of the six common chart types in our datasets, *i.e.*, (from the left) a line chart, a bar chart, a population pyramid, a heatmap on a geographic map, a scatterplot on a geographic map, and a scatterplot. We analyze them in §3.4.3.

is in terms of its color or shapes compared to the other parts of an image, which accounts for human perception of natural images. However, our results suggest that it might not be suitable for understanding human perception for a specific task.

Text Perception. Our findings are in stark contrast to the prior work [63] showing humans mostly focus on textual information in a visualization. In Scanner Deeply, the concentration of gaze on text is somewhat similar to, or sometimes even weaker than that on the shape of the chart. We hypothesize that the importance of textual information in visualization may differ by the designated task. If the task is to read visualization closely and comprehensively, textural information will carry a lot of importance. But, the overall structure of visualization becomes more critical if the task is to classify visualization types. This finding is in line with the results presented by Polatsek et al. [65].

Oftentimes, the model trained on the SALICON dataset also produces gaze predictions that focus on textual information. However, we argue that it is because the saliency is largely determined by the level of contrast in the low-level features (e.g., colors) of the text with respect to the chart background. Note that we typically use distinct colors in texts for perceptual clarity. Thus, neural networks—that are good at capturing such input differences—will learn them during training and reflect the differences in the gaze predictions.

Perception of Low-level Features. Charts in images are generally drawn so that they can stand out from the backgrounds visually. As we discussed above, this is the main reason that all three models capture low-level features. But, there is a noticeable difference between the Scanner Deeply and the baseline models. Scanner Deeply utilizes different levels of prioritization in capturing low-level features.

Suppose our model only emphasizes the low-level features that are similar to the baselines (see gaze predictions conducted by SimpleNet with Salicon dataset in Fig. 3.6). In (A), the prediction should focus on every component in the image with white color (as it has the highest contrast to the black background). However, our model focuses on a specific slide in the pie charts. It does not mean that we de-prioritize low-level features; the Scanner Deeply does prioritize low-level features when it is needed. In (C), humans need to focus on the texts at branching points to understand the sankey tree chart. In this case, our model emphasizes the texts in black, which contrasts with the yellow-colored background the most.

3.4.3 What Structure Does the Scanner Deeply Focus?

We now analyze further the gaze predictions generated by the Scanner Deeply. We specifically focus on the structures (or shapes) our model primarily focuses on visualization images. We present examples for five most popular chart types: line chart, bar chart, pie chart, geographics map, and scatterplot. We show those examples in Fig. 3.7.

Bar Charts. For bar charts in Fig. 3.7 (B), the model mainly focuses on the bars of the chart, though the gaze is focused on the center of the chart. This is not an unusual phenomenon, as chart images are generally located at the center of the image. Variants of bar charts also follow a

similar pattern. For example, in population pyramids (see Fig. 3.7 (C)), the gaze covers the entire shape of the chart.

Pie Charts. In Fig. 3.6 (A), the gaze prediction is on the center of the chart. It checks the angles of each pie and looks at the boundaries. Most pie charts have labels next to the pie, attracting the gaze. However, the concentration of gaze is usually less than that of the chart.

Line Charts. The model reads the line that passes through the chart for line charts. When the color of the line stands out from its environment, the saliency follows the line until it ends (see Fig. 3.6 (B)). When there are multiple lines in the chart, it follows all of these lines. However, if the line is similar to the background color, then it often fails to follow the line. For example, when a line is drawn on a grid plane with similar color, it confuses distinguishing between the grid and the line, and it follows both the grid and the line.

Geographic Maps. We have two types of geographic maps: (1) maps containing textual information and (2) superimposed visual components. In the first one (in Fig. 3.7 (D)), the gaze prediction focuses on the boundaries and texts with large font. In the second one (in Fig. 3.7 (E)), the prediction focuses on parts of the map with more visual components than parts with fewer components, e.g., if there are dots on the map, the gaze is focused on where the concentration is high.

Scatterplots. Generally, in scatterplots (see Fig. 3.7 (F)), our model focuses on each dot, with the highest concentration of interest in areas with the highest concentration of dots.

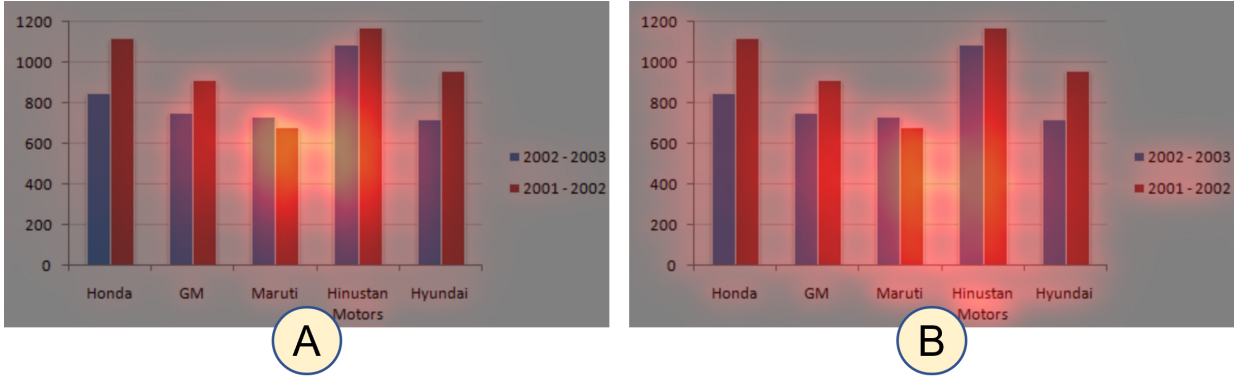


Figure 3.8: **Temporal eye movements in gaze predictions.** By training SimpleNet on datasets representing different time scale, ScannerDeeply can predict human visual perception over time. In (A), we train our model on the gaze maps collected from the first 0.7 seconds, while we train our models on the gaze maps for the entire 7 seconds. Our model predicts that human eyes will focus first on the distinct areas in charts and then focus on text components (e.g., legends) later on.

3.4.4 Analyzing Temporal Patterns in Human Perception

Our webcam-based data collection offers unique information over other datasets: we collected eye movements *over time*. We recorded eye movements every 0.05 seconds over 7 seconds in total, which allows us to analyze temporal patterns in human visual perception. Using this advantage, we evaluate an intriguing hypothesis: It has been known that humans first scan the overview of a visualization image and then perceive its details. To this end, we train our neural network on different datasets containing $0-t$ seconds, where $t \in [1, 7]$.

Fig. 3.8 illustrates the gaze predictions for an image from our model trained with 0.7s period data (left) and the model trained with the full 7s data (right). We first observe that the Scanner Deeply, trained on the 0.7s data, looks at the center of the charts. We found two reasons: (1) We made participants reset their eyes at the center; thus, they started by looking at the center. (2) We also see that to understand the chart type (i.e., bar), participants do not need to look at other information.

However, when trained with the entire data, the Scanner Deeply starts focusing on other details, such as legends or texts in the x -axis where the highest bars are. This confirms our (and also a hypothesis well-known to the visualization community) that human perceives an overview first, and then details come. Our findings also raise questions about the community’s practice in gaze predictions. Most prior work [1, 2] offers a single gaze map for each image, which may lead to missing opportunities to understand human perception in more depth.

3.5 Discussion

In this section, we discuss the implications of our work in two ways: (1) an assessment of our data collection methodology, (2) two possible approaches one can consider for gaze estimation. Finally, we conclude by discussing the limitations and plans for our future work.

3.5.1 Assessing Crowdsourced, Webcam-based Gaze

We collected a large-scale dataset of gaze fixation points on visualization images using a crowdsourced platform through general-purpose webcams. There is clearly a benefit in terms of time and cost when using a crowdsourced platform than when conducting a lab study. We spent approximately \$900 to collect around 12,000 gaze logs. This yielded approximately \$0.07 to \$0.08 per gaze log. This is about 40% more expensive than cost expectations using TurkEyes [42]. However, one must take into account that our payment was based on a targeted rate of \$15 per hour, whereas that of TurkEyes was \$10.

There was an obstacle during the experiment—the time it takes to collect sufficient amount of data. While our initial expectation of collecting period was 30 days, it took more than 50 days

to collect 10,000+ gaze logs. We attribute this to the fact that the task of collecting eye movement data on webcams is a challenge when using general-purpose webcams. Participants are required to conduct frequent calibrations, potentially multiple times, if they did not reach a particular threshold. We struggled to collect large-scale data at a fast, constant pace despite compensating at the relatively high rate of \$15 per hour.

3.5.2 A Priori or A Posteriori?

A Better Method for Gaze Estimation

Our results show that gaze patterns on visualizations are task-specific and domain-specific. This goes in line with the experiment conducted by Polatsek et al. [65]. In developing a gaze prediction model, they propose a gaze prediction model based on an *a priori* approach where the model is made as a combination of an image- and an object-based saliency model, similar to the DVS model that combines a model for natural images with a text identifier. In contrast, our work takes a *posteriori* approach that trains a neural network on empirical gaze data. Here, we discuss advantages and disadvantages for both approaches.

Compared to a prior approach, our data-driven, a posteriori approach requires a large-scale dataset and substantial computing power to train a sophisticated neural network model. However, there is no guarantee that a priori-based model will provide solid performance in the general case. For simple low-level tasks, a priori-based models can predict gaze views as well as a data-driven a posteriori model without difficulty. Issues will arise when dealing with tasks that require a priori knowledge about human perception where no such knowledge exists. This may occur when the model encounters an entirely new type of chart not covered by existing perceptual models. It

also becomes a problem when balancing between known factors is required. For example, while conventional knowledge suggests that people focus on textual information in a chart, the overview task in our experiment yielded more focus on chart shape than text. For both of these examples, a data-driven a posteriori model uses sheer computing power to learn from the large-scale dataset.

On the other hand, an a priori model for gaze prediction is built on operational and interpretable perceptual knowledge. Such knowledge can be informed by existing and ongoing research in perceptual psychology [48, 97]. The neural network implementing our a posteriori model, in contrast, is opaque and not understandable to humans [98]. Furthermore, new training datasets and models may be needed to cover all conceivable tasks, datasets, charts, and domains to guarantee robust performance for the deep learning method. Nevertheless, one important question we raise for future work is whether neural networks, such as our Scanner Deeply model presented in this paper, can teach us anything about perceptual psychology in humans.

3.5.3 Limitations and Future Work

As shown in our qualitative and quantitative experiments, gaze prediction models that are task-specific and domain-specific are desirable. Even if several works (e.g., Yabus’ [82] and Michal and Franconeri’s [83]) suggest that the task guides human attention she is conducting, most of the recent work (shown in §3.1) on predicting gaze heatmaps focuses on collecting and developing models generally working well. That being said, building such models for individual tasks may require iterative efforts. However, we highlight that our work presented a method and desirable choices for the crowdsourced data collection approach in conjunction with the low-technology requirement.

We emphasize that our paper’s datasets and methodology shed some light on interesting future work directions. As gaze prediction on visualization is an active area of study, we first envision extending our approach for various tasks as future work. We also want to highlight that the scalability issues in building prediction models for different tasks may inspire interesting future work. For example, one could build foundational models for gaze predictions, similar to models developed in computer vision (CLIP [99] or GPT-3 [100]), and then fine-tune those models for multiple downstream tasks. We further envision that exploring the feasibility of few-shot learning in this context to tackle data scarcity issues could be promising for future work.

Separately, in contrast to the datasets presented by the prior work, our dataset contains gaze dots collected over 7 seconds, which may reflect temporal human eye movements. We further envision future work using our dataset to conduct in-depth analysis of temporal eye movements to understand human perception better.

3.6 Conclusion and Future Work

In this work, we developed a Scanner Deeply, a deep learning-based gaze prediction model using chart images as the input on a specific task. After studying a taxonomy of 30 prior works, we decided to take a task-specific, crowdsourced, webcam-based eye-tracking methodology. We collected more than 12,000 chart images and conducted a crowdsourced user study to collect gaze logs. With this dataset, we trained a convolutional neural network, SimpleNet, to predict gaze heatmap when a chart image is provided as the input. In comparison with other methods, gaze heatmaps produced from our method focus primarily on the structure of the chart. From this, we deduced that predicting gaze is task- and domain-specific. Furthermore, we have shown quanti-

tative improvement in our model compared with a neural network trained on natural images and the DVS model. We evaluated the effectiveness of our approach and discussed two approaches to predict gaze.

There are still various unanswered questions in understanding how people perceive and understand visualizations. As a way to reveal a small part of this yet unknown field, we plan to extend our work by developing a gaze prediction model that uses a dashboard comprised of multiple charts. We aim to explore the dynamics behind human perception in multiple charts, how each chart plays in the sense-making processes of the users. Furthermore, we also plan to assess what neural networks are not capable of in predicting gaze views on chart images. In our experiment, we mainly focused on what neural works are capable of perceiving from gaze dots of charts. However, since these models are not humans, their abilities are not guaranteed to be a representative of the human eye [98]. We will examine the weaknesses of neural networks and other statistical models in gaze prediction on charts.

Chapter 4: Perceptual Pat: A Virtual Human Visual System for Iterative Visualization Design

Visualization design, just like any other design discipline, is inherently iterative [101]. During the course of designing a novel visualization, a designer may go through dozens or even hundreds of ideas, sketches, and prototypes before settling on a final composition. However, while experienced designers have developed the ability to view their own work through the eyes of their intended audience, it is clear that all designers would benefit from having access to an objective and unbiased audience to use as a sounding board for each design iteration. Is this label readable? Are the peaks and troughs in this line-series chart salient? Would a non-expert recognize this chart type? Being able to answer such questions at the drop of a hat would be invaluable, but fellow designers are not always available to provide feedback and empirical evaluation is unfortunately so costly in terms of time and money that it is impractical.

That is, until Pat came along. Pat is always game to take a look at a new version of your visualization, regardless of the time of day. Pat will tell you honestly what he thinks of your visualization design. In fact, Pat will be precise and give you a detailed breakdown of the colors, saliency, and legibility of your design. And, what's more, Pat doesn't sleep or take breaks. In fact, Pat can be invoked with a mere click of your mouse button.

In this paper, we present PERCEPTUAL PAT: an extensible suite of image processing, com-

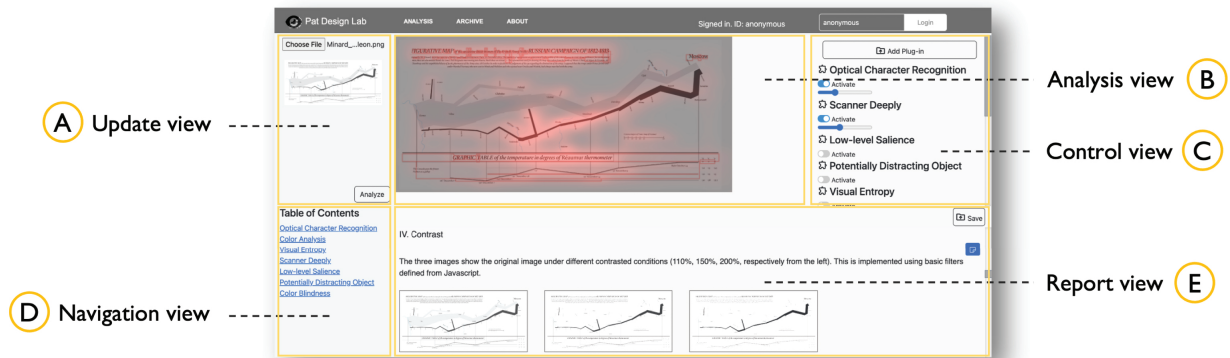


Figure 4.1: **Pat Design Lab.** Applying Perceptual Pat to Charles Minard’s famous Napoleon’s March to Russia diagram. The Pat Design Lab is the web-based interface to the Perceptual Pat suite of perceptually-inspired image filters that can be used by a designer to receive rapid and inexpensive feedback on a visualization being designed.

puter vision, and machine learning models that taken together forms a virtual human visual system suitable for supporting visualization design and evaluation. While we obviously do not intend for Perceptual Pat to be a drop-in replacement for a real human by a long shot, the Pat suite provides a collection of perceptually-inspired image filters that combine to yield a comprehensive picture of what a person would see when viewing a visualization. Examples of such filters include the virtual gaze maps, such as the Scanner Deeply virtual eyetracker [6], color vision deficiency filters, text legibility and optical character recognition (OCR) scanning, color analysis, etc. Inspired by the virtual human known as “Virtual Jack” [102] (or just “Jack”), which has long been a staple in ergonomics design, Perceptual Pat is a complement and not a replacement for human testing. Instead, the purpose of Pat is to provide easy access to multiple rounds of quick and cheap feedback before a design is evaluated in a focus group, expert review, or usability study.

At its core, the Perceptual Pat suite is held together by the PAT DESIGN LAB, a web-based software system for managing iterative visualization design. Using the Design Lab, a designer can upload evolving versions of a design over time, each time running the Perceptual Pat tests and

receiving an interactive report in response. The Design Lab allows for analyzing and studying the output of the tests as overlays display on top of the visualization itself. The tool also has functionality for designers to add their own annotation and notes to the report. Finally, different reports can be interactively compared using the tool, enabling the designer to see how their design has evolved over time. Furthermore, the Pat suite is based on a flexible plugin architecture, enabling the Design Lab to be easily extended with third-party image filters.

We validated the Perceptual Pat suite and the Pat Design Lab in a longitudinal user study involving 4 expert visualization designers using the tool for their own design project. We first met with our participants in individual meetings to introduce the tool and its functionality, and then again a week later at the conclusion of their design project. During this time, the participants were asked to design and refine a new visualization from scratch using the tool. Our analysis of the resulting reports and user annotations indicate that a majority of causes for change in the design were attributed to our system. During the exit interviews, participants acknowledged the effectiveness of the Perceptual Pat in detecting problems within visualizations, as well as its convenience in providing design feedback.

The contributions of this work are:

- The concept of a *virtual human visual system* (VHVS) as a suite of models providing feedback on a visualization image indicative of human perception;
- The Perceptual Pat implementation of a virtual human visual system for iterative visualization design;
- The Pat Design Lab for supporting iterative evaluation of visualization design using the Perceptual Pat suite;

- Qualitative findings from a longitudinal study involving four expert designers using our prototype implementation in their own design projects; and
- Findings from external evaluators who assessed the design process and outcomes from the longitudinal study.

4.1 Background

Here we provide an overview of the background for our work on the Perceptual Pat suite by first explaining how visualization is a design discipline and reviewing systems that scaffold the design process. We then review work in vision science and visualization that shows how understanding perception can also support visualization design.

4.1.1 The Role of Feedback in Visualizations

Data visualization is still a largely empirical research field, and iterative design is thus a key component in authoring a new visualization [101]. It has also entailed a focus within the field on design heuristics and rules of thumb, such as Edward Tufte’s reviews of effective visualizations [103], visualization textbooks with a design emphasis [101], and many blog posts and practical handbooks drawn from practice.

All design benefits from external feedback. There are two main mechanisms for feedback: user and usage feedback vs. peer and supervisor feedback. The former is more common in academia and focuses on validation, often through empirical evaluation. Munzner presents a nested model for visual design and validation [104] that incorporates primarily the former through various forms of validation. She presents four design layers for creating visualizations: (1) do-

main problem characterization, (2) data/operation abstraction design, (3) encoding/interaction technique design, and finally (4) algorithm design. Sedlmair et al. [105] develop a nine-stage design study framework (consisting of three higher level category), and provide practical guidance for conducting design studies. The three high-level categories are: (1) personal (precondition), (2) inward-facing (core), and (3) outward-facing (analysis). Peer feedback, on the other hand, is more common in industry and design practice, where critique is critical when authoring visualizations intended for mass consumption. It has also reached academia within the general interaction design community. Bardzell et al. [106, 107] talk about the importance of criticisms in designing interfaces, and assert that they raise people’s perceptual ability, which over time constitutes a heightened sensibility or competence.

Comparison. Obtaining external feedback requires time and money and is not always accessible. In contrast, Perceptual Pat provides external feedback using automated methods, and is thus quick, always available, and virtually cost-free.

4.1.2 Facilitating the Chart Design Process

External feedback helps to improve chart design, but there exist many different ways to provide such feedback. One method is based on practical visualization recommendation. Building on seminal automatic visualization work by Jock Mackinlay [108], Tableau’s Show Me [109] feature recommends charts based on data types as well as best practices. Since then, various types of recommendation tools have been developed, such as those based on data properties [12, 110], perceptual principles [12], expert feedback [111], large-scale dataset-visualization pairs [15], and design knowledge [13].

Another method is to directly aid iterative design for visualization. The data visualization saliency (DVS) [1] model is a general-purpose saliency heatmap generator for visualizations. This saliency map enables designers to understand a viewer’s attention on a chart. Recently, there is also a rising interest in techniques that provide automated design feedback on visualizations using a linting or sanity check metaphor. Examples are tools for detecting chart construction errors [18], visualization mirages [16], and deceptively-designed line charts [112]. Finally, VizLinter [19] even provides solutions to help chart designers.

Comparison. Our Perceptual Pat suite is similar to many of these techniques, but it does not recommend visualizations to be authored, leaving authoring entirely in the hands of the designer. Instead, our approach is to provide a toolbox of different filters that together can give the user multiple lenses through which to view the visualization artifact they are designing. This is akin to a supertool [113] augmenting the capabilities of the designer a hundredfold.

4.1.3 Understanding Human Perception for Visualization

Understanding how humans perceive charts is vital to supporting design, and has been an active area of interest within vision science. These efforts began as early as the end of the 19th Century in work done by the so-called “Berlin School” of experimental psychology. This eventually led to the development of *Gestalt psychology* [114], a theory of mind based on a holistic view of human visual perception where the sum of the perceived “gestalt” is qualitatively different than its component parts, and in effect has an identity of its own.

Experimental work within vision science has also spent significant effort collecting empirical data on how humans perceive charts. In 1926, Eells et al. [115] study how people viewed

statistical graphics. Croxton et al. [116] compare bar charts with pie charts in 1927 and study the effectiveness other shapes for comparison in 1932 [117]. In 1954, Peterson et al. [118] perform experiments for eight different statistical graphics. These findings, and many more, were summarized in Cleveland and McGill’s seminal 1984 paper [119] on graphical perception of statistical graphics.

Empirical work on sophisticated visualization mechanics has continued at a rapid pace; an exhaustive survey is beyond the scope of this paper. Bateman [5] study the impact of visual embellishment compared to minimalistic chart design, finding memorability improvements. Chalbi et al. [120] extend the original Gestalt laws for dynamic graphical properties. Michal and Franconeri [83] present findings on the order readers follow when viewing visualizations (e.g., looking at the tallest bar when reading bar charts). Many of these detailed empirical findings have been condensed into specific design rules in a recent paper by Franconeri et al. [121], listing both visualizations that succeed in effectively communicating data through visualizations and those that fail to do so because of illusion and misperception.

With the introduction of high-performance eye tracking devices, researchers use eye movement data [96] to understand human perception. This provides us with new knowledge about what causes confusion in charts [55], visual patterns that benefit recall [40], visual saliency as a measure of attention [94], and assessment methods for visualization proficiency [122]. It is even possible to infer one’s personality with eye movement patterns during chart reading [123].

Comparison. All this work are candidates for inclusion into our pragmatic take on a virtual human visual system.

4.1.4 Modeling Human Perception

Our work is inspired by the concept of *Jack*, a human simulation system devised by Badler et al. [102] in 1999. Jack is an abstracted version of a human body, with special focus on the body’s physical properties and its movements. In general, simulating humans as virtual agents provides a virtual experience that can be used to detect and prevent many of problems in a pre-emptive manner during early design.

We propose an approach for modeling not the physical body of a human, but its visual system for the purposes of supporting visualization design. In 1993, Lohse propose the first simulation of a human perceptual system based on a cognitive modeling approach drawn from both experimental and theoretical findings up to that date [124]. More recently, Haehn et al. [125] study the use of convolutional neural networks (CNNs) as a possible candidate model for a human perceptual system. While they were able to replicate several of Cleveland and McGill’s seminal graphical perception results, they ultimately decide that CNNs are currently not a good model for human perception. Finally, Shin et al. [6] use crowdsourced eyetracking data to build a bespoke deep learning model that simulates human eye movement to generate gazemaps on any uploaded data visualization presented to it.

Comparison. Perceptual Pat is different from all of this prior work because it integrates and synthesizes many models based on image processing and computer vision. In fact, many of the aforementioned tools have already been integrated into the Perceptual Pat suite. However, we are aware of no design-oriented virtual human visual system similar to ours.

4.2 Design: Virtual Human Visual System

We informally define the concept of a *virtual human visual system* (VHVS) as a simulated human perceptual system based on computer software filters that use imagery as their main input channel and outputs information about the expected human visual response to this input. In our implementation, we think of these image filters as *perceptually-inspired* to indicate the pragmatic and practical approach we adopt in this paper; instead of attempting a high fidelity simulation of the human visual system, our goal is to provide actionable information to a designer iterating on a visualization artifact.

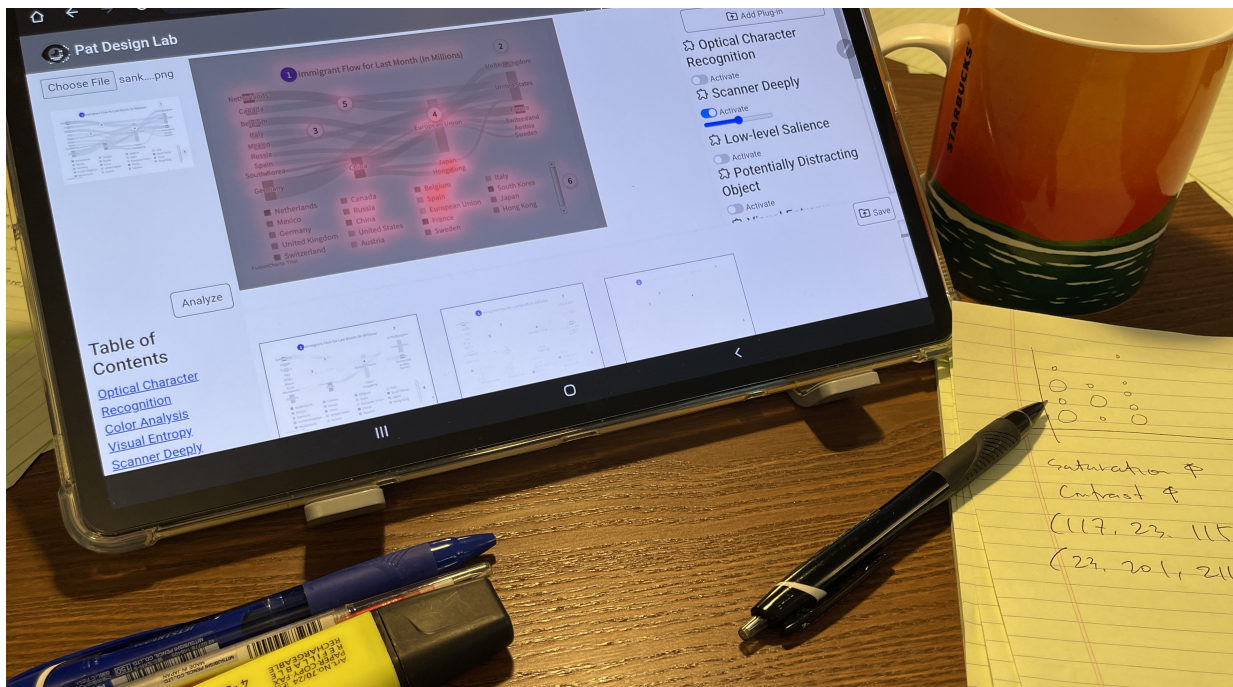


Figure 4.2: **The Pat Design Lab in action.** Example of a designer using the Pat Design Lab during their chart design process.

4.2.1 Motivation

There are many potential reasons for designing a virtual human visual system (VHVS), ranging from vision science—e.g., the ability to completely simulate the human visual system at high fidelity—to more practical applications for specific design and evaluation situations. Our goal in this paper is the latter: while the overall VHVS goal may seem lofty, we are primarily interested in providing pragmatic and actionable feedback to designers through a suite of image filters “inspired by” human perception (see Fig. 4.2 for a demonstration of its intended use). In other words, we do not purport to faithfully model the human visual system, but rather to pragmatically support iterative visualization design.

4.2.2 Scope

The scope of our VHVS implementation is to serve as a *supertool* [113] that uses AI and computer vision to augment, amplify, and extend the capabilities of a human designer during iterative visualization refinement. While our intended user is a visualization designer of any skill level, we note that our approach is not currently to suggest fixes to identified concerns. This means that while the feedback is useful to anyone, an expert designer will often be in a better situation to address it because of their greater experience. We organize the external design feedback into two main types:

- **Design feedback and critique:** This form of external feedback involves receiving criticism—preferably constructive, i.e., focused on improvement, rather than merely pointing out flaws—from peer designers or supervisors. Designers often work in teams or at least as part of organizations with multiple designers, so critique is intrinsic to design [107]. How-

ever, receiving feedback from peers can be time-consuming because (a) the process is laborious in itself, and (b) peers are often not immediately available because of their own commitments. Furthermore, feedback from an uninitiated peer can often be general and not sufficiently detailed. Nevertheless, design feedback remains important, not only for designers working alone on a project, but also for teams who can benefit from an outside and unbiased critical eye on their work.

- **Empirical evaluation:** Given that visualization is a primarily empirical discipline, evaluation involving human participants is essential [126]. This is true for both academia as well as industry and practice [127, 128]. However, even small-scale empirical evaluation studies are costly in terms of time, money, and preparation—they certainly cannot provide answers to specific questions designers have about their visual design at short notice. For this reason, empirical evaluation is conducted at a time granularity of days or, more likely, weeks.

4.2.3 Design Requirements

Because of the automated nature of the VHVS that we are proposing, it is important to note that we are not replacing but *augmenting* human feedback. This means that the fidelity requirements are lower because our automated feedback is meant for guidance rather than enforcing specific perceptual or design rules. We summarize our design requirements from the motivation and scope above as follows:

- DR1 **Rapid:** To facilitate a conversation between the designer and the VHVS, feedback must be rapid, preferably yielding output in less than a minute or two (the faster, the better).

DR2 **Inexpensive:** Similarly, to promote frequent iteration, the feedback cannot require costly investment, or ideally any investment at all. This likely precludes crowdsourced critique, which still incurs some cost.

DR3 **Automated:** The feedback should be mostly automated and not require intricate configuration or setup; ideally, the user should be able to submit the current state of their visualization artifact.

DR4 **Progressive:** In recognition of the iterative refinement commonplace in visualization design, the feedback should track the evolution of an artifact over time.

DR5 **Constructive:** The feedback provided should be organized to help the designer improve their artifact.

4.2.4 Practical Perceptual Feedback

Focusing on pragmatic, actionable, and practical perceptual feedback on a visualization artifact frees us from having to design a fully comprehensive and accurate model of the human visual system. Instead we choose types of feedback that will aid the designer in making improvements to a visualization that will directly benefit viewers. We summarize the main types of such feedback below. Note that this is a suggestion and not an exhaustive list.

⚙️ **Visual saliency:** Saliency is a measure of the patterns and parts of a visual scene that attracts a viewer’s eyes [94], and has long been a core part of vision science [44, 70, 71]. However, visual saliency is also a highly practical and pragmatic aspect of visualization design in helping the designer determine which parts of a visualization will attract the

viewer and in what order, and, analogously, which parts will not. Receiving unbiased and objective feedback on the visual saliency can certainly help a designer iterate on their visualization to ensure that the saliency is consistent with their intent.

- *Eyetracking*: Eye trackers empirically measure saliency by collecting eye movement data from the real world [39]. Eyetracking technology is now becoming cheaper and increasingly available, enabling designers to use themselves as test subjects. However, such hardware solutions are beyond the scope of our work.
- *Virtual eye trackers*: A virtual eye tracker is trained on eye movement data to generate artificial gaze maps, enabling the designer to conduct a virtual eye tracking experiment. Large-scale eye movement data can be used to build eye movement prediction models, such as the work by Itti et al. [44], the CAT2000 benchmark [74], and SAL-ICON [2]. While gaze prediction models can be used for this purpose, the Scanner Deeply [6] tracker is specifically trained on visualization images.
- *Information theory*: Information-theoretic approaches to visualization [129] measure the ratio of raw data that are communicated using a visual representation. Calculating the local entropy across a visualization image can thus be seen as a theoretical measure of its visual saliency; its information content.

🎨 **Color perception:** Color is a basic building block of any image, data visualization in particular [101, 130]. Effective use of color is therefore a key factor in any design projects involving data visualization.

- *Color statistics*: Understanding dominant color schemes, color distribution, and color

scale choices is a useful mechanism for any visualization designer.

- *Color choice:* Visualization practice stipulates using a limited number of distinguishable and easily named colors [131], potentially as a function of the mark used [132].
- *Opponent processing:* Opponent process theory casts color perception as balances between three pairs of colors [133, 134]; while the exact constituent colors are disputed, these are often held to be red vs. green, blue vs. yellow, and black vs. white. Pragmatic visualization would promote avoiding color combinations that involve both parts drawn from one of these pairs.

A Text: Most visualizations incorporate text in some form, much of it central to understanding the scale, extents, names, and details of the visualized data. Textual characters are obviously visual objects that are modeled by other vision models, but because of their special meaning in visualization, text is worth studying on its own. We propose several specialized forms of text identification and classification feedback.

- *Legends:* Some visualizations require legends to enable deciphering color allocations or color scales. Identifying the legend, or notifying the designer that none is present, would therefore be useful feedback.
- *Labels:* Axis labels, titles, and tick marks are central to comprehending a visualization.
- *Textual content:* Disregarding the visual representation of the text, what about its actual content? Feedback on spelling, grammar, and meaning can help the designer here.

Visual representation: Some perceptual feedback may be specific to the visualization technique being used. While some of this type of feedback may stray into visualization linting [18, 19] for finding chart design and construction errors, we here focus our feedback on perceptual aspects.

- *Chart type:* Merely using an automatic classifier to identify the chart type can be useful feedback to a designer. If the designer is working on a non-standard visualization and the classifier does not recognize it, this may be indicative to change representation, or to better signpost the representation. Alternatively, if the designer is using a standard chart type and it is not recognized (or incorrectly so), this may be a signal that they need to improve and standardize their visual design.
- *Data extraction:* Taking chart recognition a step further would be to use automatic methods to recover the data from a visualization, essentially reverse-engineering the visualization [135–137]. This would enable the designer to determine if the visual encoding is lossy by recovering symbolic data that has been encoded.
- *Visual embellishment (“chart junk”):* Visual embellishments to charts—sometimes referred to as “chart junk” [103]—while potentially beneficial to the memorability of a chart [5], may detract from the chart or even distract the viewer compared to a more minimalistic visual representation [138]. Automated object recognition can inform the designer about any visual embellishments and their potential for distraction.

Vision science: Beyond the visual saliency discussed above, there are many useful metrics from vision science that we may pragmatically adopt for visualization design feedback. As stated above, since our goal is not to accurately model the visual system, we can instead

choose concepts that lend themselves to iterative design.

- *Preattentiveness*: Preattentive features [139–141] are those that guide the viewer’s attention so that they “pop out” in a visual scene, and that cannot be decomposed into simpler features. Automatically detecting and highlighting preattentive features in a visualization would be highly useful, because they can help the designer confirm conscious design choices and discover—and likely address—inadvertent ones.
- *Ensemble processing*: How do people estimate characteristics from a group of visual objects, such as marks in a scatterplot? *Ensemble processing* [142] models how the visual system computes averages of visual features in a group with even complex shapes and configurations. Implementing an ensemble processing filter could help designers understand how groups of visual marks would likely be interpreted by the viewer.
- *Shape identification*: Recognizing and identifying shapes in a visualization artifact may be another confirmatory piece of feedback for a designer. If shapes are not correctly identified, perhaps due to scale or overplotting, the designer may use this feedback to make revisions to their artifact.
- *Image segmentation*: More of a computer vision than a vision science tool, *image segmentation* [143, 144] is the process of partitioning an image into segments or regions based on image content. These segments may help a designer to understand about the fundamental structure of the visualization artifact being designed.
- *Moving object tracking*: While our approach in this paper is based on static screen-shots of a visualization, a dynamic animation may yield further perceptual informa-

tion about a visualization. In particular, providing feedback on moving objects may be useful given human perceptual limits on tracking multiple objects [145, 146]. It could also be used to understand temporal aspects of an animation [147], which can aid perception.

🦿 Accessibility: The accessibility of visualizations has recently become an area of burgeoning research within the community [135, 148], and many visualization designers are in dire need of assistance in making their charts accessible. Here we review some specific forms of feedback that can help.

- *Color vision deficiency:* An estimated 300 million people in the world suffer from some form of color vision deficiency where their ability to distinguish colors is diminished. There are already websites and native tools to show what a visualization tool looks like depending on the form of color vision deficiency; integrating such feedback into a framework would help further. In addition, Angerbauer et al. [149] present several findings drawn from a large-scale crowdsourced assessment of color vision deficiency in visualization that could be operationalized for a virtual human visual system.
- *General accessibility:* The Chartability framework by Elavsky et al. [150] provides heuristics for evaluating and auditing visualizations based on their accessibility to people with different disabilities, including visual, motor, vestibular, neurological, and cognitive ones. However, the framework must currently be applied manually during an accessibility audit. Operationalizing these heuristics into an automated model would enable integrating it into a framework such as ours.

4.3 The Perceptual Pat Suite

Perceptual Pat is an implementation for a virtual human visual system designed for iterative design of data visualization artifacts. In this section, we describe its system architecture, components, and implementation.

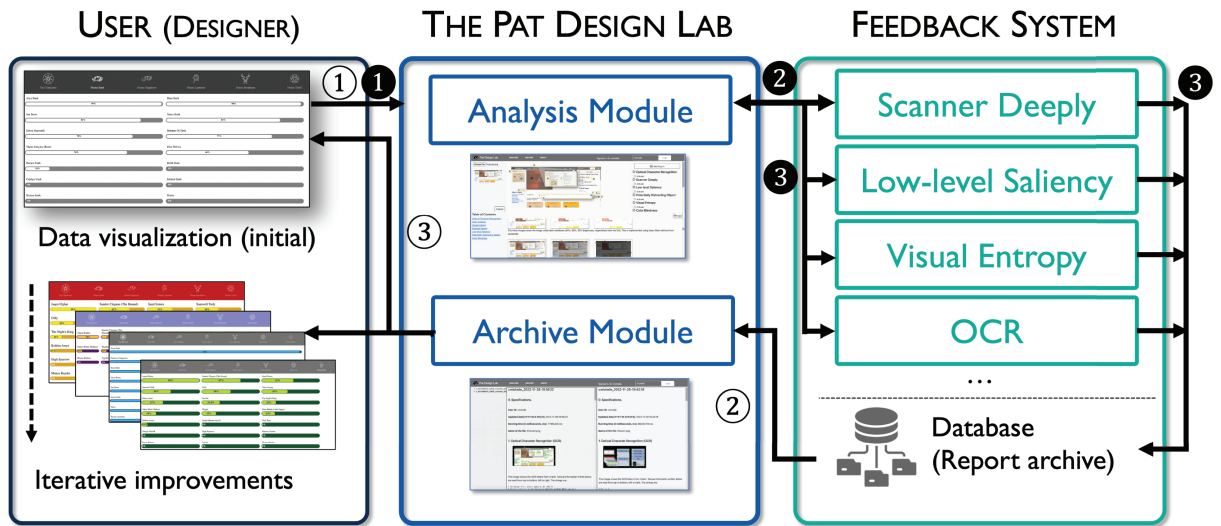


Figure 4.3: **System overview.** The Perceptual Pat suite is comprised of a user interface (The Pat Design Lab) and a feedback system. The user interface receives a data visualization (chart) from a user and automatically performs an analysis of the chart. The feedback system consists of diverse visual analysis components and uses them to generate a report containing design feedback. (We used Perceptual Pat to iteratively refine this figure by providing a freelance graphic designer with the original version and a PDL report.)

4.3.1 System Overview

The Perceptual Pat suite is designed to meet the requirements in § 4.2.3 (see Fig. 4.3 for an overview; the black ❶ and white ❷ circles refer to the steps in the figure). The suite consists of two components: a web-based user interface (the Pat Design Lab) and a feedback system. The user interface (PDL) contains analysis and archive modules—see the following subsection for

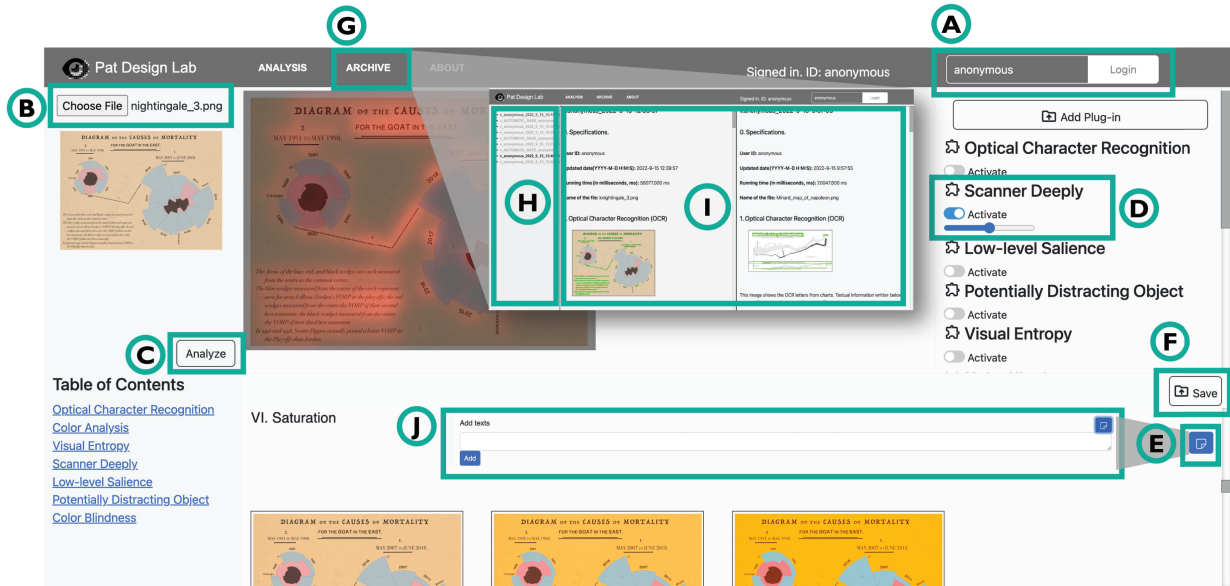


Figure 4.4: **Designing visualizations with the Pat Design Lab.** The user can log in to the system by signing in on (A). Then, they can upload images on (B), and start the analysis process by clicking the ‘analyze’ button in (C). They can choose filters in the control view, and at the same time control the opacity of the filter in (D). To take notes in the report, they need to click the blue button in (E). Then, they can add texts in (J) and click the button ‘add.’ Users have to manually press the save button in (F) to keep record of them report with notes. They can check her past versions of the design by clicking the hyperlink ‘archive’ at the navigation bar (G). In the archive view, they can select a version of her interest in (H), and the most-recently chosen version will appear at the left window, pushing the old one into the right window (I).

more details. ❶ The analysis module receives a chart image from a user in a graphics format (e.g., .png, .jpeg, or .jpg). ❷ This action automatically triggers the feedback system to generate a report containing the results of the diverse visual analysis we develop in § 4.3.3 (DR1, DR2 and DR3). ❸ The report will be shown to the user, and is also stored in the database for future reference.

The archive module is designed to support the iterative chart design process of a user (DR4 and DR5). ❶ It allows a user to retrieve the report that they generated before or to compare two or more reports at the same time. The user will have a list of report names shown in chronological order. By clicking those report names, the user interface requests all reports from the feedback

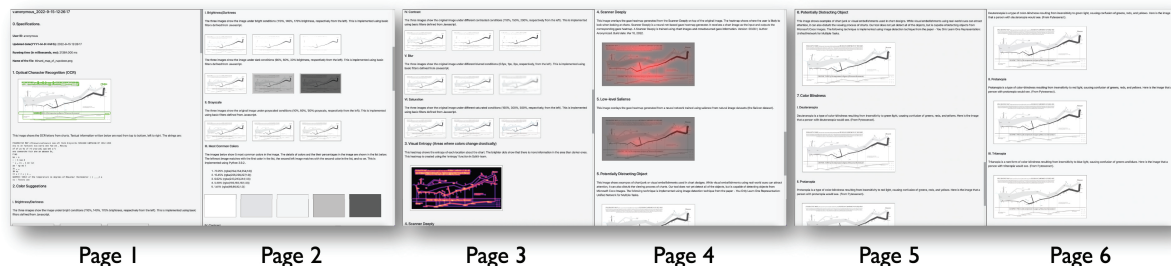


Figure 4.5: **Report generated by the Pat Design Lab.** The report contains information about filters used for the analysis of the updated chart images. It consists of 8 sections. These include (1) the chart's specifications, (2) OCR, (3) visual entropy, (4) a Scanner Deeply, (5) low-level salience, (6) potentially distracting objects (Chartjunk [5]), and (7) color-vision deficiency.

system. ② The feedback system will pull those reports from the database. ③ Reports are shown in a single view side-by-side and are scrolled up/down together.

4.3.2 The Pat Design Lab

The Pat Design Lab (PDL) is a web-based single-page application. At its core, PDL consists of three functions: upload, analyze, and save. Users can upload a screenshot of their chart image. Then, they can analyze the image using the perceptual components provided by the PDL. Finally, users can save their versions for comparison with past designs.

To initialize the Pat Design Lab, the user needs to login to the system (Fig. 4.4(A)). Once they have signed in, Pat Design Lab activates and the update view (Fig. 4.1(A)) appears. In the update view, they can choose a file to upload into the main interface. They can start the analysis process by pressing the 'analyze' button in Fig. 4.4(C).

When PDL completes the analysis, the analysis report is displayed in the report view (Fig. 4.1(E)). Users can toggle filters that are overlaid on top of the visualization image, as shown in Fig. 4.4. The filter opacity can be controlled using a slider (Fig. 4.4(D)). Multiple filters can be overlaid at once to enable studying compound effects.

The report view in Fig. 4.1(E) provides access to the full analysis report; Fig. 4.5 shows an example. The component names in the navigation view provides easy access to corresponding report § 4.1(D). Users can also add notes about each component for each section by clicking the document button in Fig. 4.4(E), which displays a text input interface (Fig. 4.4(J)). They can save the version with the notes by clicking the blue-colored button with a document icon in Fig. 4.4(F).

The archives tab provides access to saved reports (Fig. 4.4(G)). This page provides a full list of past reports and notes (Fig. 4.4(H)). It also shows two windows that lets users compare two reports of their choice (Fig. 4.4(I)).

COMPONENT NAME	FEEDBACK TYPE	IMPLEMENTATION
Scanner Deeply	🔦 Visual Saliency	Scanner Deeply [6]
Low-level Saliency	🔦 Visual Saliency	Trained SimpleNet [43] using Salicon dataset [2]
Visual Entropy	🔦 Visual Saliency	SciPy library [151]
Color Suggestions	👁️ Color perception	CSS filters, Python Image, OpenCV libraries
OCR	📄 Text	Google PyTesseract-OCR [152, 153]
Chartjunk/Visual Embellishment	📊 Visual Representation	YoloR [154]
Color Vision Deficiency	👁️ Accessibility	Python Color-blindness library [155]

Table 4.1: **Components composing Pat’s feedback system.** Names, feedback type and implementation of seven components included in the suite. These were selected to be representative and cover the design space in § 4.2.4. We provide additional technical and performance details on these components in Appendix (supplemental material).

4.3.3 Current Perceptual Pat Components

Table 4.1 shows the components in Pat’s feedback system, drawn from § 4.2.4. The focus of this work is not primarily in evaluating the performance of these components, but in exploring how access to a virtual human visual system can help users during iterative design. For this reason, our goal was not to exhaustively cover the entire design space in § 4.2.4, but to find a representative sample of components.

Towards this end, we chose several perceptual components that cover a wide range of the feedback types in § 4.2.4 and are readily adaptable from existing computer vision and machine learning libraries. Note that Pat uses a plugin system, so adding new components is straightforward. Below we explain how we implemented the seven components.

Optical Character Recognition (OCR) Optical character recognition is a computer vision technique [156] that detects text characters from natural images. This technique gives the designer an “smoke test” of the legibility of the text; if the OCR technique fails to detect the text, there may be a legibility problem and redesign may be needed.

A Scanner Deeply Scanner Deeply [6] is a *virtual eyetracker*; a gaze heatmap generator using a neural network model trained on more than 10K instances of eye movement data with chart images as the visual stimuli. The model will generate a simulated heatmap predicting where a person’s attention will be directed when viewing a visualization.

Low-level Saliency This saliency heatmap generator uses a neural network model trained on the Salicon dataset [2], which contains fixation maps for a natural image dataset (i.e., Microsoft

COCO [157]). The component is particularly useful in showing viewer attention for visualizations that contain natural image data, or in visualizations that are situated within the world, such as for a visualization embedded in Augmented Reality.

Visual Entropy This component generates a heatmap showing *visual entropy*; pixels whose RGB values differ from neighboring ones. This can potentially give designers awareness of the data distribution in the image.

Color Vision Deficiency This filter component provides color overlays to enable the designer to see how people suffering from different types of color vision deficiency (CVD) would see the visualization. The component supports three types of CVD: (1) deuteranopia, (2) protanopia, and (3) tritanopia.

Chartjunk/Visual Embellishment Our component for detecting chart junk is implemented using the YoloR object detection algorithm [154, 158], which detects real-world objects visible in the visualization image. The model is limited by its training, which may cause it to fail to detect all objects, but this functionality can still serve as early warning.

Color Suggestions Our color suggestions component is implemented using CSS filters and include blur, gamma, grayscale, contrast, and saturation. It enables envisioning how a visualization would look under different color themes.

4.3.4 Implementation Notes

The Perceptual Pat suite is implemented as a client/server web application. The Pat Design Lab was built using HTML, CSS, JavaScript, and JQuery. We implemented the feedback system using the Python Flask web framework¹; the analysis components were implemented server-side in Python 3. We store data in MongoDB.² During the user study, the platform was hosted on a Ubuntu Linux server equipped with an Intel i7-12800K processor, 32GB RAM, 2 Pascal Titan RTX GPUs, and 2TB of flash memory.

4.4 User Study

We conducted a user study to assess the utility of the Perceptual Pat suite and the web-based Pat Design Lab implementation. The study asked participants to use the Pat Design Lab in support of a visualization design task spanning between three and five days. Rather than compare our approach to an existing tool, our study is mostly qualitative in nature and focuses on understanding how professional visualization designers might use the tool to iteratively refine a visualization artifact. We motivate this choice by the fact that there exists no directly comparable tool that would serve as a useful baseline.

Here we describe the participants, methods, and metrics of this study. In the next section, we report on the results.

¹<https://palletsprojects.com/p/flask/>

²<https://www.mongodb.com/>

ID	GENDER	AGE	EDUCATION	JOB TITLE	YEARS IN VIS
P1	Male	28	Master's Degree	Software Engineer	More than 3 years
P2	Female	27	Master's Degree	Ph.D. Student in Visualization	More than 3 years
P3	Female	43	Master's Degree	UX/UI/Visualization Designer	More than 5 years
P4	Female	24	Bachelor's Degree	Freelance Visualization Designer	More than 1 year
P5	Male*	–	(no information)	(no information)	(no information)

Table 4.2: **User study demographics.** Our participants were recruited from the Data Visualization Society Slack channel (<https://www.datavisualizationonsociety.org/slack-community>) and community of IT Engineers in Facebook South Korea. * = participant P5 did not finish the study.

4.4.1 Participants

We recruited 5 professional visualization designers using the Data Visualization Society Slack channel. Table 4.2 presents an overview of the participant demographics. All participants were paid a total of \$40 as an Amazon gift card upon completing the study (or its equivalent in the participant’s requested currency). Unfortunately, one participant abandoned the study after a week without producing any design artifacts; their demographic data is shown in Table 4.2, but in no other part of the paper.

4.4.2 Apparatus

The study was conducted online in its entirety using the participant’s own computers. We imposed no specific hardware on the study, only that the participant would use a personal computer, and not a mobile device such as a smartphone or tablet for the design task. We recommended them to use the website in Google Chrome. Since the Pat Design Lab is designed to use only screenshots of data visualization tools, participants were free to use their own preferred visualization tool and workflows.

4.4.3 Design Task

Participants were asked to do only one thing during the 3-5 days of the study: to design a new visualization from scratch using the Pat Design Lab as supporting software. The visualization could be anything: use any dataset, any representation, and for any use (including merely for the purpose of this study). They were asked to record at least 5 versions of their visualization into the Design Lab. Ideally the five stored versions would be taken from representative stages in

the design process. Participants were instructed to read the reports, write at least one annotation, and to endeavor to use the reports to improve their designs.

As stated above, we placed no restriction on the design workflow or visualization tools used in the process—participants could use whatever tool they preferred, or even switch tools during the process. For example, a participant could start with a pen-based sketch and then proceed to using Tableau, Excel, or Spotfire. We only stipulated that the participant spent a total of at least **two hours** on the design process.

In summary, we made the following requirements on the visualization design process:

- Resolution minimum 400×300 , maximum 1000×1000 pixels in resolution (higher resolutions were downscaled).
- At least 5 versions stored into the Pat Design Lab.
- Artifact must include at least one chart (any form of chart), and could include multiple charts.
- Artifacts may include text.
- Artifacts may **not** include photographs.
- No confidential or identifying information or data is to be included.
- Participants give permission for publishing images of their created artifacts in academic papers about the work.

4.4.4 Procedure

Our study was approved by our university's IRB. The study consisted of three phases that spanned over three to five days, followed by an independent fourth phase involving external evaluators. We scheduled the dates for each phase during initial recruitment. The fourth phase was independent of the preceding three and did not involve the designer participants. Below we describe each of the four phases in detail.

Phase I: Initial Interview. During our initial interview (Phase I), we gave a brief introduction of the study and then asked participants to provide informed consent (signature waived due to the online format) using an online form according to stipulations from our IRB. We then collected participants demographics using another form. The remainder of the initial interview consisted of training during which time the experiment administrator (the first author of this paper) demonstrated how to create an account in the Pat Design Lab, authenticate and log in, and then upload a screenshot of a visualization into the tool. The administrator then showed how to run a Perceptual Pat analysis and interpret the resulting report, as well as how to compare two different reports.

After finishing demonstrations, the experimenter asked the participant to repeat each step, and answered any questions that the participant had. Then the experimenter gave the participant their design charge, including the requirements listed above. The participant was given the design task in electronic form. Finally, the session ended by confirming the date and time for the exit interview (Phase III). Each session lasted approximately 30 minutes.

Phase II: Individual Design Process. During the intervening time between the initial interview (Phase I) and the exit interview (Phase III), the design process (Phase II) consisted of the

participant working on the visualization artifact they were designing. Participants were free to spend this time in any way they wished—we only asked that they spent at least a total of **two hours** on the design work and that they used the Pat Design Lab to support the process. They were instructed to reach out to the experimenter with any questions or problems that arose during the study; none of them did, and the system was stable during the time period of the experiment.

Phase III: Exit Interview. The exit interview (Phase III) involved asking the participant about their experiences using the Pat Design Lab, their feedback about its strengths and weaknesses, and their design process using the tool. The experiment administrator then stepped through the participant’s version history in the Pat Design Lab, one version at a time and asking about details for each version. In case the participant had stored more than five versions, the experimenter asked the participant to identify the five most significant versions. Each session lasted approximately 30 minutes.

Phase IV: External Assessment. Finally, we recruited three external evaluators to objectively assess the visualization design process. All evaluators were senior visualization faculty or researchers with experience in teaching data visualization and/or designing their own visualizations. They gave informed consent and were then given access to the sequence of visualization versions for each of the participants. They were then asked to provide their assessment of the quality of changes for each pair of versions (i.e., from version 1 to 2, 2 to 3, etc) as well as from the initial (version 1) to the final (version 5) design using a 1-5 Likert scale (1 = significant decline in quality, 3 = neutral, 5 = significant improvement in quality). They were also asked to motivate their assessment using free text.

4.4.5 Data Collection

Interviews were video and audio recorded, and the audio was transcribed for later analysis. Furthermore, demographics, subjective preferences, and tool rankings were collected using online forms. The Pat Design Lab itself was the main source of data collection. This data encompasses each of the visualization versions uploaded into the Pat Design Lab by the participants, including the screenshot of the visualization, the resulting Pat design report, and the annotations added by the participant. These annotations were augmented with the spoken comments that participants made about each version during the version walkthrough in Phase III.

While we included all versions uploaded by participants in our analysis, we also asked participants to identify the five most significant versions in case they had uploaded more than five.

4.5 Results

Here we report on the results from our user study. First, we introduce participants' responses on interviews prior to starting the experiment. Then, we analyze the evolution of the visualization artifacts and the impact of Perceptual Pat on the iterative design. Finally, we present participants' comments after conducting the experiment.

4.5.1 Results from the Initial Interview

Besides overall instructions and a demographics survey during the initial interview in Phase I, we also asked the participants three questions to understand how they usually get feedback for visualizations; see Table 4.3 for the questions and responses. As is clear from the table, the

majority of participants receive feedback from people who are easily reachable, such as peers, colleagues, or supervisors (P2, P3). If the visualization is upon a client's request, then these designers tend to get feedback directly from clients to confirm that their work is acceptable, as it is the most direct way to understand the client's intentions (P1, P4).

P#	Q1: How do you currently get feedback to improve a visualization?	Q2: Are you familiar with any tools that provide design feedback?	Q3: If such a tool existed, would you find it useful?
P1	clients	not familiar	somewhat useful
P2	peers, friends	not familiar	very useful
P3	supervisors, managers, peers	not familiar	somewhat useful
P4	client, peers	not familiar	somewhat useful

Table 4.3: **Initial interview with participants.** Responses for Q1 are given in the participant's own order.

When asked whether they were familiar with tools that provide design feedback, no participant was familiar with any such tool. All participants felt, however, that such a tool would be useful. As a main reason for this, all participants pointed out that receiving timely feedback is sometimes a challenge and there are only a few people who can provide feedback in visualizations. Asking the same people over and over can be problematic. P2 provided us another reason—that it is difficult to verbally communicate about visualizations with peers and by using a machine, she might not have such a problem.

P3 did raise an issue about the credibility and transparency of the system. She felt that such a tool would be useful, but said, *“I am not sure if I can 100% trust the feedback provided by a machine. If the machine makes a comment, I would like to know why, in detail, it provided that comment, so that I can decide whether or not to trust it.”*

4.5.2 Evolution of Visualization Snapshots

Fig. 4.6 shows how visualizations of the four participants evolved over 5 iterations of designs. Here we describe how each participant changed their design based entirely on the uploaded images. In the following subsection we explore the rationale behind each change.

Participant 1. Originally, P1 created his chart using Matplotlib, a Python library for drawing charts. The first version of the chart was the initial version that Matplotlib creates, without any manual constraints involved. In the second and third versions, he enlarged the fonts of the letters within in the heatmap, the title, and axis labels. In the fourth version, he changed the heatmap from a bipolar to a unipolar color scheme. At last, he removed the potentially disturbing icons next to the labels, to keep from disrupting the readability of the chart.

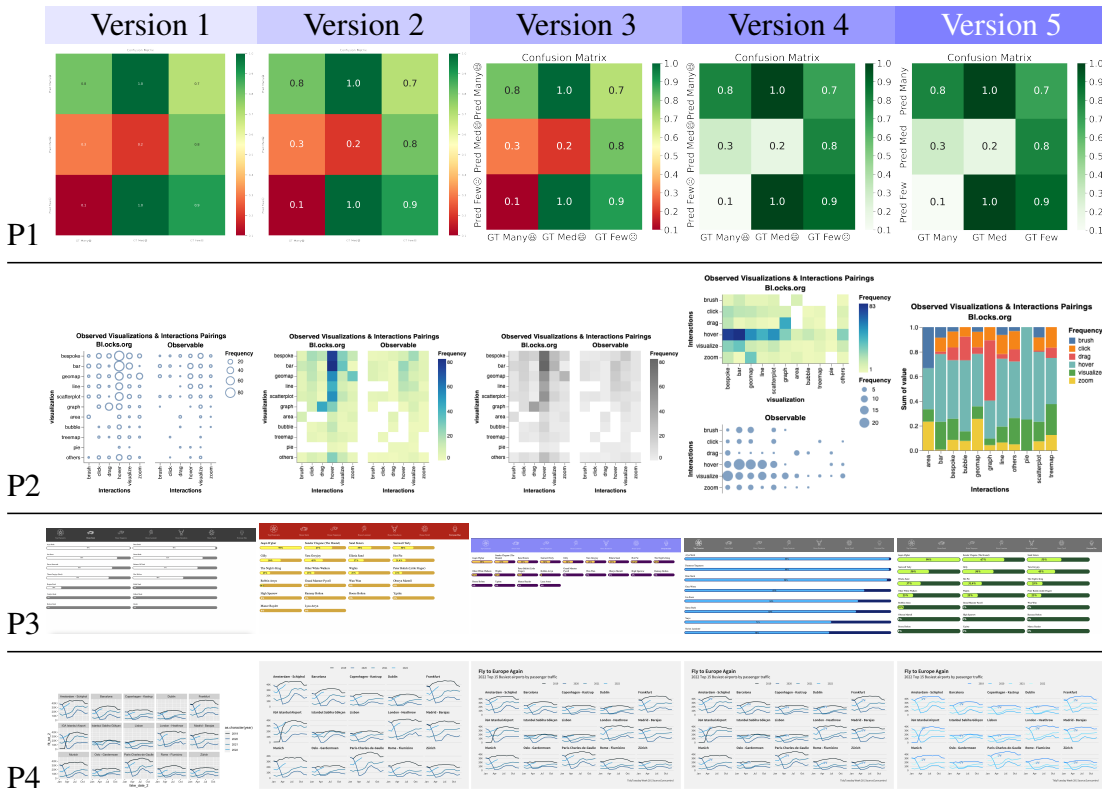


Figure 4.6: Overview of collected designs. These images represent the five most significant versions (as identified by the participants) during the design process of creating a new visualization artifact from scratch using the Pat Design Lab and the Perceptual Pat Suite. All four participants from the study are represented (five were recruited, but one person abandoned the study after a week).

Participant 2. P2 selected a research dataset classifying the interactions commonly used for specific chart types. Initially, she started off with a 2-axis grid scatterplot (version 1), where the horizontal axis represents the types of interactions (e.g., hover, brush, click, etc.), and the vertical axis represents the types of charts (e.g., bar, line, etc.). The size of the blue-outlined circle in each point refers to the frequency of charts. In the next version, she changed her chart into a heatmap (version 2), and then into a grayscale (version 3). Then, in the fourth version, she created both a heatmap and a grid scatterplot. This time, she filled the circle with blue. Finally, in the fifth version, she created a stacked bar chart, where each column represents the proportion of different interactions and interactions are represented as different colors of the bar.

Participant 3. P3 used a dataset on characters in a video game. She used D3 to create a visualization. She started by creating a bar chart with 2 columns using grayscale (version 1). Afterwards, she tried, in an iterative manner, different designs by changing the number of columns, sizes of bars, and color (versions 2, 3, 4). She finalized her visualization using green-colored bars on a gray background (version 5).

Participant 4. P4 wanted to visualize air passenger traffic data of 15 different cities. She decided immediately to use line charts displayed in a 3 by 5 grid format. The initial version had gray charts with a white background and a legend at the right side of the visualization. In the second version, she moved the legend to the top and removed grids that are drawn in each chart. In the next version, she added a title in black, subtitle in dark gray, and source information about the chart at the right end of the chart. In the fourth version, she added a small airplane embellishment to the image. In her final version, she performed finishing touches by changing the color of the chart to a more saturated blue.

4.5.3 Evolution Rationale

Recall that in the report we asked the participants to record what influenced them to refine their design. These descriptions, intentions, and reasons for the change are based on both these annotations as well as the interviews conducted with each participant.

Participant 1. P1 was conducting feature analysis of a deep neural network, and wanted to create a chart to share with his peers in the company. He initially looked at the OCR to see if the component could detect letters in the chart. He consistently checked the size of the font in

the chart, and left notes below the report on OCR. For example, he left a note below the report on OCR in the third version, as “*Increased the font size, after knowing that the label size of the chart is too small.*” Then he noted that the pink-colored areas in the heatmap were not visible for people with deuteranopia, and that the icons next to the label were detected by the potentially distracting object. This led him in the last version to change from pink and green color palettes to green palettes, and to removed small icons next to the labels.

Participant 2. P2 wanted to construct a visualization that is capable of effectively contrasting the different types of interfaces used for each chart. She was interested in searching for the chart that can best show the types of interactions per each type of chart. In the first version, she noted that she was not satisfied with using size as the only encoding type. Then, in her second version, she changed her scatterplot into a heatmap. She noted two points: (1) increasing the brightness past a certain threshold can lead to information loss, and (2) the visual entropy was overly focused towards the labels. In her third version, she converted the chart into a grayscale version and found that the grayscale version did not have colorblind issues. In her fourth version, she added two different types of charts to the design. She was also concerned about the results of the Scanner Deeply, noting that gazemaps focused on textual information such as labels and titles when she wanted people to focus more on the data representation. However, she noted that while this concerned her, she was unsure whether this was a good sign or not. Finally, in the last version, she replaced the two charts with one stacked bar chart with each bar representing the proportion of different interaction per each chart. Then she noted that (1) it did not make sense to have the data encoded as either points or a square in a heatmap, and that (2) the Scanner Deeply finally focused on the data representation more than the labels and legends.

Participant 3. P3's goal in creating the visualization was to find the proper size of columns for the bar chart and the proper color. In the initial version, she left notes on all filters except the color vision deficiency component. In the second version, she changed the color using red palette, and created 4 columns to present bars from 2 columns. Here, she discussed the visual entropy and low-level salience results, noting that there is more visual entropy because of the icons in the navigation bar. She was also concerned that the low-level salience concentrated on specific parts of the bars, but was unsure how to change the design. In the third and fourth versions, she tried to change the colors and the number of columns in the main view. In the third version, she noted that the new low saturated colors helped reduce user attention and instead focus on the view with bars. In the fourth version, she noted that this version looked like the worst design, considering the results from visual entropy, Scanner Deeply, and low-level salience. In the final version, she used these findings to change to a highly saturated green to represent bars and to darken the navigation bar.

Participant 4. After receiving a report on his initial version, P4 left remarks about all components in the report. In particular, she noted that the Scanner Deeply focuses only on the center of the chart, and that the grids installed in each chart raised the complexity of the charts. Consequently, she removed grids in each chart, and placed the legend at the top of the charts in the next version. Again, she left a note that the attention from the Scanner Deeply is still focused too much on the center of the visualization. So in the third version, she added a title, subtitle, and caption to the chart to see if they help divert the attention to other parts of the chart. On the report from the third version, she made comments on OCR and Scanner Deeply. Using OCR, she identified that title, subtitle and captions are noticed by the OCR component, and also that

the attention of the chart diverged after putting in title and subtitle. In the fourth version, she slightly modified the heights of each chart and added a small airplane icon in the chart. Here, she mentioned that image with higher saturation improved the visibility of line charts. At last, in the final version, she changed the colors of the lines with more saturated blue. From this, she observed that the increased saturation also helped diverge the attention of gaze heatmaps from the center to charts in various locations.

4.5.4 External Assessment

Table 4.4 summarizes the quality change ratings of the external evaluators in the final phase (Phase IV) of our experiment. As can be seen from the results, the evaluators mostly felt that all four participants had successfully improved on their visualization designs over the course of the design process. Only P2's design received an average rating below the neutral; evaluators were split in their assessment of this design process.

The evaluators also provided qualitative feedback on the changes they saw. Even if the evaluators only saw the visualization designs (and not the Perceptual Pat reports), their written feedback often called out specific design improvements, such as more salient color scales, more legible fonts, and better visual layouts that seemed inspired by Perceptual Pat feedback. For example, they noted several instances of increased legibility, larger font sizes for labels, improved color scales, better visual layout, and better visual encodings. We were able to match the majority of these qualitative observations to the feedback that the individual participants received in the longitudinal user study.

PARTICIPANT	v1 → v2	v2 → v3	v3 → v4	v4 → v5	OVERALL
P1	4.50 (0.71)	3.50 (0.71)	3.50 (2.12)	4.50 (0.71)	4.50 (0.71)
P2	4.50 (0.71)	2.00 (1.41)	2.00 (0.00)	3.00 (2.83)	2.50 (2.12)
P3	3.00 (0.00)	3.00 (0.00)	3.00 (0.00)	4.00 (1.41)	3.50 (0.71)
P4	4.00 (0.00)	4.33 (0.58)	3.33 (0.58)	2.33 (0.58)	3.67 (0.58)
All	4.00 (0.71)	3.21 (0.98)	2.96 (0.67)	3.46 (0.98)	3.54 (0.82)

Table 4.4: **External assessments.** Average 1-5 Likert scale ratings (1 = significant decline, 3 = neutral, 5 = significant improvement) by external evaluators assessing the quality of changes from one version to another ($v_x \rightarrow v_y$) as well as overall from the initial to the final version. Standard deviations are given within parentheses.

4.5.5 Post-experimental Interview

After the experiment, we asked the participants six questions with regards to their experience in Perceptual Pat. Q1, Q4, Q5, and Q6 are shown in Table 4.5, and Q2 and Q3 are described in Table 4.6.

P#	Q1: How useful was the Pat Design Lab?	Q4: What tool(s) was the most useful?	Q5: What tool(s) was the least useful?	Q6: What tool would you most like to add?
P1	very useful	OCR, color vision deficiency	chart junk, color suggestions	suggestion tool
P2	somewhat useful	Scanner Deeply, color vision deficiency, visual entropy	color suggestions (ex-cept contrast)	suggestion tool, result interpreter
P3	somewhat useful	Scanner Deeply, low-level salience	color suggestions	suggestion tool
P4	somewhat useful	Scanner Deeply, OCR	color suggestions (ex-cept saturation)	suggestion tool, result interpreter

Table 4.5: **Post-experimental interview with participants.** After each participant finished updating their versions of visualization, in the post-experimental interview we asked 6 following questions to understand how they get feedback and whether they are familiar with a tool that provides design feedback. In Q1, the answers are written in the order each participant mentioned.

Q2: Advantages of Perceptual Pat	Q3: Disadvantages of Perceptual Pat
Detects problems within a chart (P1, P2, P3, P4)	No guidance for improvement (P1, P2, P3, P4)
Convenience in asking for feedback (P1, P2, P3, P4)	Difficulty in interpreting results (P2, P3)
Feedback on general audience reaction (P2, P3, P4)	Unclear target audience (P3)
Helps overcome biases in visualizations (P2, P3)	
Acts as a checklist for improving visualizations (P1, P4)	

Table 4.6: **Advantages and disadvantages of Perceptual Pat.** After the experiment, we asked participants about the advantages/disadvantages of a virtual human visual system providing visual feedback for visualization designers. The table shows a summary of the answers from 4 visualization designers that participated in our experiment.

When asked again about whether the tool was useful, all of them answered positively about the system, answering either ‘somewhat useful’ or ‘very useful.’ Then we asked for the advantages as well as the disadvantages of Perceptual Pat. As for advantages, all participants agreed that the tool is capable of detecting problems within the chart. We could identify from their notes in the report that many changes in the design were based on these reports. Another advantage shared among all participants was that it is convenient to obtain feedback compared to when asking to get feedback from clients, peers, and supervisors. P2 and P4 liked that it provides feedback quickly without having to ask peers for feedback. P3 thought it could thus save time in getting feedback. Three participants liked the fact that tools such as Scanner Deeply provide feedback on how general audience would react to their visualizations. Also, both P2 and P3 thought that the tool help overcome their biases in visualizations. For example, P3 originally thought that high levels of saturation would make the chart look bad aesthetically, but when changing the saturation value higher using Pat Design Lab, she felt that adding saturation to the visualization was a choice worth considering. Finally, another advantage P1 and P4 pointed out was that the tools work as a checklist in evaluating the design, including factors that are otherwise easily overlooked (e.g., inclusive designs for the color blind).

In total, participants listed three disadvantages about the Perceptual Pat. All participants were looking for not just detecting problems, but actual guidance on how to improve a chart. While the filters provided can help detect problems in the design, it is up to the designers to find the right solution to the problem. The participants thought that a guidance could facilitate the process of their design processes. Also, P2 and P3 expressed concerns about the difficulty in interpreting the results. P2 said that while some feedback is easy to address, some of the filters are not easy to interpret. She said, “*some of the feedback, such as changing contrast or brightness*

is easy, but some aren't. Consider the Scanner Deeply, for example. It is sometimes difficult to judge whether the heatmap is a good or bad.” P3 also talked about the importance of interpreting the tool. To use Perceptual Pat in industry, she said, *“one must be fully comfortable with the mechanisms behind how the tool works, so that when something is wrong, we know how to fix it.”* Last but not least, P3 asserted that visualizations are designed differently by different target audience, and the tool will become more useful if it suggested a specific group of components according to the target audience.

In Q4 and Q5, when asked about the most/least useful component, three participants thought that the Scanner Deeply was the most useful component, being followed by OCR and color vision deficiency. All participants thought that the color analysis and suggestions were the least useful. As for which new component people would most like to add (Q6), all participants asked for a functionality that could provide design suggestions. In addition, P2 and P4 thought that a component that can provide interpretation about the result could also help in improving a visualization design.

4.6 Discussion

We here discuss the findings and implications from our user study, followed by the limitations of our work, as well as our plans for future work.

4.6.1 Benefits of the Perceptual Pat Suite

Shorten the feedback loop. All the participants commented on the convenience of getting immediate feedback from Perceptual Pat. In their regular workflows, participants reported their

common practice for receiving design feedback was to ask their peers or supervisors, which is time-consuming and resource-intensive. In contrast, Pat gives them to receive feedback within a minute or two, thus saving time and resources for actual design activities. This confirms our rationale for embarking on this project in the first place.

Provide design guidance. Results from our user study confirmed that the Pat Suite indeed helped our user study participants to improve their visualization design. The quality assessments by the external evaluators show a positive improvement trajectory for each of the four participants even if there were inevitable setbacks during the process.

The Perceptual Pat feedback was seen as direct and actionable. P3, for example, mentioned that the gaze map predicted by Scanner Deeply caused them to avoid users mostly focusing on the center of their visualization. To tackle this issue, they raised the saturation of the chart. After changing the colors to those with high saturation, they re-ran the analysis and found that the Scanner Deeply predicted that the user’s attention would be more equally distributed across the whole chart. Based on notes left in their reports, we also found that that the majority of improvements—such as increasing font size, changing the color scale, or reorganizing the spatial layout—were made based on the feedback received from Pat.

4.6.2 Limitations and Future Work

Interpretability. Perceptual Pat’s feedback system contains components that utilize deep neural networks for generating feedback, such as Scanner Deeply or chart junk detection. The decision-making process of those models, while accurate, is opaque, and therefore developing *interpretable* models and tools is an active area of research [159–162]. P3 pointed out that for

designers to be able to trust tools such as Perceptual Pat requires significant knowledge of how those models work. Despite our best effort in providing the technical details of Scanner Deeply, P3 wanted to know more about the factors that led to the model’s feedback responses. Said P3, *“In the end, it is the designer who decides whether to take advantage of this tool or judge whether what I am looking at is a real problem. I must be able to know the extent I can trust when I see a tool so that I can properly judge.”* Thus, we hope to improve the Perceptual Pat suite in the future by explaining the design feedback it generates.

Recommendations. Perceptual Pat can be seen as an early-warning sanity checker for visualization designs (i.e., a perception-based visualization linter [19]) similar to the unit-tests that software engineers use to ensure the functionality of the software they develop. However, just like unit-tests, while Pat provides a list of potential issues for a specific visualization, it does not offer recommendations for how to fix them. P1 said *“I know my design has several issues, but Pat does not show suggestions for fixes.”* P2 pointed out by saying that *“I love this approach, but it would have been better if Pat could tell me ‘these are the issues and here is how you can fix them.’”* Participants also mentioned that, for less experienced designers, giving concrete guidance would reduce the duration of the design process. However, experienced visualization designers may not need such guidance, as they generally know how to refine their designs from Pat’s feedback. Thus, a future research direction is to add design recommendations to the Perceptual Pat suite.

Accuracy. As with any automated method, there is always a risk for erroneous results. False positives can be particularly problematic, since it may give the designer the impression that ev-

everything is fine, and thus not investigate further. A false negative, on the other hand, means that the designer will inspect the problem manually. This should only be a problem if the number of false negatives flagged is excessive.

As a case in point, applying an OCR filter to extract all of the textual labels in a chart may yield some recognition failures that a human would not make. That is acceptable: the OCR filter is acting as a canary in a coal mine by indicating possible concerns that may not be a problem in practice, or which may only be a problem for people with diminished visual acuity. For this reason, Perceptual Pat has been designed with the philosophy of favoring false negatives over false positives; it is better to highlight many potential problems than to run the risk of missing a real one.

Bias. Similarly, automation may also give rise to bias. Paradoxically, the fact that Perceptual Pat currently does not provide design recommendations may actually reduce this effect. On the other hand, even the mere reporting of perceptual aspects cannot be said to be entirely unbiased; for example, highlighting one type of issue (visual attention, text legibility, or color saturation) and not other types (such as animation, use of visual channels, or Gestalt Law groupings; all examples of components not currently supported by Perceptual Pat) means that the former issues will tend to get highlighted and thus fixed, whereas the latter ones won't. In other words, Perceptual Pat clearly has blind spots (no pun) that will have an impact on visualization designs iterated using the tool. The only remedy—besides progressively adding new components to the suite to eliminate each of these gaps—is to at least make users aware of their existence. Thus, in our future deployments of the tool, we will inform users not only which perceptual errors the suite checks for, but also which ones it does *not*.

Generality vs. Specificity. While all the participants in our study felt that Perceptual Pat’s feedback helped improve their designs, some pointed out that this “one-size-fits-all” strategy may not be effective for all audiences. Said P3, *“it would be helpful if Pat would suggest not only which chart to use, but also which tool to use depending on the audience and domain.”* In fact, we see the potential for taking this idea to its limit by providing customized “flavors” of Pat—or Steve, Susan, and Xian—that embody specific audiences, domains, and visual design philosophies. We leave such integration and customization as future work, however.

4.7 Conclusion

We have presented Perceptual Pat, a virtual human visual system designed for iterative visualization design. Pat comprises a suite of image filters that are built using computer vision and related technologies to provide design feedback akin to what a designer may receive from both peer designers and supervisors as well as perceptual evaluation results collected from an empirical usability evaluation. The Pat Design Lab provides a web-based interface to Perceptual Pat, enabling a user to track the evolution of their design work by repeatedly uploading new screenshots of their work, analyzing it with Pat, and viewing the results in a design report. To assess the utility of Pat and his Design Lab, we conducted a longitudinal evaluation involving four professional visualization designers who used the tool to support the design of a new visualization artifact. Our findings showcase the utility of receiving quick turnaround feedback from the Pat suite in the design process.

Chapter 5: Visualization: Automating Design Feedback for Visualization Designers Using Large Language Models

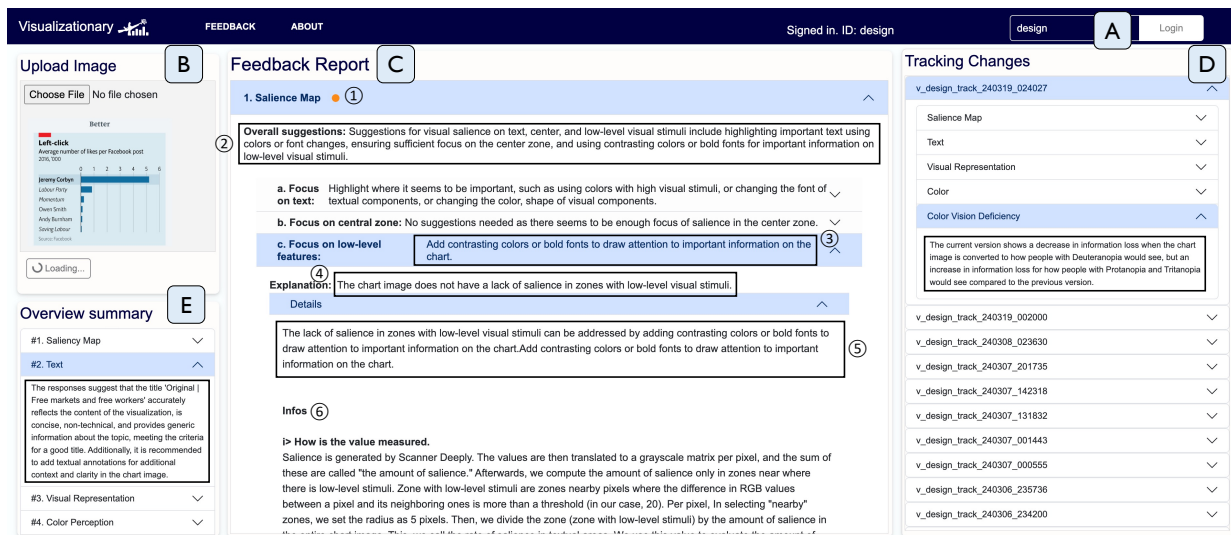


Figure 5.1: **Interactive report.** Report on a visualization artifact based on our novel analyze-clarify-guide-track (ACGT) framework. It is generated by automatically analyzing (“analyze”) the artifact using an extensible collection of image filters grouped into categories: salience, text, visual representation, color, and accessibility. After reporting results from each image filter (center column), the report then interprets (“clarify”) the findings using natural language understandable to a novice visualization designer. This is followed by natural language suggestions (“guide”) on how to address weaknesses in the design. Finally, the right side of the report shows changes over time (“track”) as the designer iteratively refines the artifact. The texts in black boxes are responses generated by ChatGPT.

To democratize the transformative power of data in today’s information society, it is not sufficient that people get access to visualizations created by others; they must also be empowered to author their *own* visualizations. As a result, recent years have seen an influx of interactive

tools for creating visualizations without writing code, such as iVisDesigner [163], Data Illustrator [164] and Data Animator [165], Charticulator [166], and Lyra [167] and Lyra 2 [168]. However, no matter how advanced these tools become, a fundamental barrier remains: authoring effective visualizations is a complex task requiring visual design skills, aesthetics, and experience that not all would-be designers possess.

Fortunately, visualization designers have nothing to lose but their chains. In this paper, we propose VISUALIZATIONARY, a tool to seize the means of visualization design by using off-the-shelf Large Language Models (LLMs) such as ChatGPT or LLaMA. The basic idea behind Perceptual Pat is to investigate how such an LLM can be used to aid the iterative design process of a visualization artifact in a workflow that we call *analyze-clarify-guide-track* (ACGT):¹

1. **Analyze** the state of a visualization using automated perceptual filters, yielding a corresponding analysis report;
2. **Clarify** the results from the perceptual filters in the report such that it is understandable to even novice visualization designers;
3. **Guide** the designer in making changes that address identified concerns in their visualization; and
4. **Track** the trajectory of the visualization artifact over time during its iterative design process.

We report on a web-based implementation of the Visualizationary system using the OpenAI ChatGPT API, showcasing how a “vanilla” LLM can be utilized for this highly complex

¹Incidentally also the initials of four of the five canonical nucleobases; more on this in § 5.2.

visualization design task. In particular, we translate chart images into text using a VLM (vision-language model) [169] and then generate automatic design guidance using an LLM as a template and text processor. The tool presents the guidance in an interactive report organized into a hierarchical structure that enables the designer to drill into the findings to the desired level of detail. We also present findings from a longitudinal study conducted with 6 novice, 4 intermediate, and 3 expert visualization designers over a period spanning a few days. During the study, each designer worked with Visualizationary to create a new visualization from scratch. We report on four successive revisions of their visualization as well as their quantitative and qualitative feedback. While Visualizationary and the ACGT workflow is primarily designed for novices, our findings show how the approach can be useful even for an experienced designer by summarizing feedback in an easily navigated interactive report format.

5.1 Background

Here we present the state of the art in the intersection of three topics: design feedback, automating visualization design feedback, and LLMs.

5.1.1 Design Feedback

Feedback is essential in all design disciplines, and it typically comes from two sources: user and usage feedback versus peer and supervisor feedback. In academic contexts, the former, which emphasizes validation through empirical evaluation, is more prevalent. Munzner’s nested model for visual design and validation [104] primarily incorporates this type of feedback. Conversely, peer feedback holds greater prevalence in industry and design practice, particularly for designs

aimed at widespread use. Nevertheless, this methodology has also permeated academic circles, notably within interaction design [106, 107].

For feedback to be effective, it must not only be understandable by the intended recipient, but it must also provide enough information so that the recipient can act upon the feedback (if necessary). What Munzner calls low-level feedback [101] is focused on fine-grained insights and improvements, and is often actionable because of its low abstraction level: e.g., “*this red color is a poor choice for a color-blind person,*” “*this bar must be made larger,*” and “*this text uses a font with poor legibility.*” However, higher-level design feedback in the form of critique, qualitative evaluation, and abstract metrics can be much harder to understand, let alone act upon. For example, learning that your visualization suffers from “poor contrast,” “poor color choices,” or is “not aesthetically pleasing” can be challenging to address.

5.1.2 Automatic Design Feedback on Visualizations

At its core, data visualization is very much a design discipline and still relies heavily on empirical methods to iteratively refine an artifact over time [101]. This has led to a significant focus on design guidelines and rules of thumb [103, 121], textbooks with a design focus [101], and numerous resources derived from real-world practice [170]. However, to personalize such design guidelines and complement personal tutoring, recent work in the research area has begun to provide various mechanisms for automating design feedback for the designer.

One feedback approach is focused on *linting*, a technique drawn from compiler design where source code is statically analyzed to flag poor coding practices (e.g., no indentation, poor identifier naming, and lack of variable initialization) that often cause errors. Analogously,

visualization linting can detect chart construction errors [18], visualization mirages [16], and deceptively-designed line charts [112]. For example, VizLinter [19] helps rectify problems in Vega-Lite specifications.

Another method uses a form of recommendation to automatically generate charts based on the specific dataset to visualize. Mackinlay’s Ph.D. work on automatic visualization [108] was one of the first endeavors in this area, and it was later realized in production as Tableau’s Show Me feature [109]. More recent visualization recommendation methods have been proposed that generate charts using data properties [12, 110, 171], perceptual principles [12], expert feedback [111], large-scale dataset-visualization pairs [15], and design knowledge [13].

Accessibility is a specific concern that is seeing emerging interest in the visualization field [172, 173]. Heuristic frameworks on general accessibility such as Chartability [150] could conceivably be operationalized to support automatic accessibility auditing. For color vision deficiency, Angerbauer et al. [149] present results from a large-scale crowdsourced assessment that could be similarly automated.

Finally, quantitative feedback can also help a designer improve their visualization. Vis-Lab [174] provides a web-based system facilitating the collection of such quantitative feedback from crowdworkers that can be used to evaluate and refine a visualization in online experiments. Perceptual Pat [7] tracks the evolution of a visualization artifact over time, assembling a report consisting of several perceptual metrics for each iteration using computer vision and related methods. In comparison to this system, our work goes beyond merely displaying perceptual metrics about a visualization artifact to explaining the findings and then suggesting how the user may address shortcomings.

5.1.3 Large Language Models

A *language model* is a statistical model that predicts the likelihood of word sequences in natural language based on a large corpus of textual training data. Recent work [175–177] shows that the primary discriminator in accomplishing complex tasks using language models is their *scale*. In light of this trend, there has been a growing number of works on so-called “large” language models (LLMs) [175, 178–181]. LLMs have trillions of model parameters, and thus, training these models is computationally demanding. It needs a large training dataset, GPU hardware acceleration, and a machine equipped with them. This means that training an LLM is expensive in time and resources.

A standard paradigm for addressing the training costs is to pre-train LLMs on a large corpus of datasets and then allow users to fine-tune these models on a set of desired tasks, e.g., question answering, language translation, and text summarization [175, 181–185]. In pre-training, a model is typically trained to do next-word prediction, i.e., given a prompt, the model returns a sequence of word tokens likely to complete the prompt. This process is performed by third-parties such as OpenAI, Google, or Meta, and such *pre-trained models* are offered to users as a form of internet service, e.g., ChatGPT [178], or as-is, e.g., open-source models such as Vicuna [179] or LLaMA [180]. In particular, services such as ChatGPT offer a personalized experience by separating sessions for each user and fine-tuning the pre-trained model based on the user’s inputs. Recent models further support *multi-modality*, with images and text being combined, and enable them to answer questions, e.g., in data visualizations: “How much does X outperform Y?” [186, 187]

Our work leverages both types of pre-trained LLMs to generate feedback: (1) Multi-modal

LLMs that take a user’s data visualization and questions on the visualization as input and return the answers to the questions, and (2) uni-modal LLMs (or standard ones) that receive text inputs and return the corresponding outputs. We use multi-model LLMs to generate textual descriptions for a user’s visualization, and we then combine the descriptions and the queries as the input to uni-modal LLMs, asking the model answer queries as design feedback.

5.2 Model: Automated Visualization Design Feedback

We propose an automated visualization design feedback mechanism for supporting iterative design of a visualization artifact. The basic idea is for the designer to upload consecutive versions of a visualization artifact as they iterate on the design. The automated system will then provide natural language feedback on the artifact to facilitate improvement.

5.2.1 Target Audience

Our work in this paper primarily targets **novice** or **intermediate designers** who do not have the required experience or expertise to critique their own work with an objective eye, or the ability to make the required changes to address problems. In other words, our automated design feedback mechanism is not intended to be a replacement for a trained designer’s critical eye. However, because design critique is critical during any design process, automated design feedback can be useful even for **experts** [7]. Similarly, automation is no replacement for a human peer or mentor to give feedback and suggestions to the novice designer. However, mentors or peers may not always be available.

5.2.2 Rationale

Based on Munzner’s visualization design framework and nested evaluation framework [101, 104], interaction design critique [107], and rich literature on guidelines for visualization design [13, 121, 166], we can identify several benefits for automating visualization design feedback:

- **Efficiency:** An automated mechanism can provide immediate feedback to support rapid iterations in the design process.
- **Consistency:** Automated feedback systems apply consistent evaluation criteria, reducing subjectivity and bias as well as human error in assessments. Human feedback, especially if collected from different sources, may be less consistent and sometimes even conflicting.
- **Scalability:** Compared to a human evaluator, an automated design feedback mechanism can handle a large volume of evaluations simultaneously, making it feasible to assess multiple design variations.
- **Inexpensive:** Automating feedback can reduce (but not eliminate) the need for engaging human evaluators, which can be costly.
- **Real-time:** An automated visualization feedback system can provide data in real-time, enabling designers to make immediate adjustments and improvements during the design process.

5.2.3 Analyze-Clarify-Guide-Track (ACGT) Workflow

We propose an automated visualization design workflow that we call *analyze-clarify-guide-track* (ACGT).² It has the following components:

²ACGT is also an acronym for adenine, cytosine, guanine, and thymine, the four nucleobases that form the building blocks in nucleic acids such as deoxyribonucleic acid (DNA). DNA, obviously, is the carrier of the genetic code for all known living organisms and many viruses. There is a joke on the building blocks of visualizations in there somewhere, but, sadly, it escapes us.

1. **Analyze:** To provide written feedback to the designer, we first need to collect textual data from the visualization screenshot; i.e., we must make the transition from images to text. We do this using automated vision filters, such as color contrast, visual hierarchy, and layout consistency (see § 5.2.4). The outcome of this analysis is a report with metrics drawn from the visualization artifact.
2. **Clarify:** Interpreting visualization metrics on their own can be challenging [7], let alone to use as a basis for revisions. The *clarify* phase translates results from the perceptual filters into understandable terms, thus bridging the gap between technical analysis and practical design improvements. It breaks down complex issues identified in the analysis into layman’s terms, explaining why certain changes are needed and what specific aspects need attention.
3. **Guide:** It is not enough to know the problem; we must also know how to fix it. Our mechanism provides such practical guidance to the designer as natural language on how to address the concerns identified during the analysis. This guidance may include recommendations, actionable insights, or best practices for improvements.
4. **Track:** Understanding the overall trajectory of the design process is important to avoid local optima. The tracking component monitors the progress of the design over time during iterative refinement. It maintains a historical record of design iterations, changes implemented, and corresponding improvements in visualization quality. This tracking allows designers to see the evolution of their work, understand the impact of changes made based on earlier recommendations, and make informed decisions for future iterations.

5.2.4 Visualization Filters

We list visualization filters used to extract perceptual characteristics from visualizations, with the goal of providing pragmatic and actionable design feedback. These filters are a suite of computer vision and vision-language models (VLMs) that we collectively call *visualization filters*. Note that the list is extensible and not exhaustive.

⚙️ **Virtual Eyetracker** refers to the extent to which areas within a scene capture the attention of an observer [94].

A Textual characters are visual objects that are used for labels, titles, and legends in visualization [188].

📊 **Visual representations** can generate feedback specific to the technique, such as visualization linting [18, 19].

🎨 **Color perception** is a basic building block of data visualization [101, 130]. Visualization practice stipulates using a limited number of distinguishable and easily named colors [131], potentially as a function of the mark used [132].

♿️ **Accessibility** is emerging as an important new topic in the visualization community [135, 148, 172, 173]. Automated feedback can be highly useful here, even for seasoned designers.

LLMs are capable of generating human-like natural language responses under various tasks. To that end, we first extract numerical metrics from a visualization and pass it to an LLM. We also instill design knowledge in the LLM so that it can guide designers. We explain this idea in the next section.

5.3 The Visualizationary Tool

Perceptual Pat is a web-based visualization design feedback system that automates natural language critique in iterative design for visualization designers. In this section, we describe its system architecture, overall workflow, and core components. We also describe the prompts we define for generating quality feedback from a LLM.

5.3.1 System Overview

Visualizationary is a web-based client-server system that tracks a visualization artifact throughout its design process. The workflow (Fig. 5.2) is based on the user—a visualization designer—iteratively uploading a screenshot of the current state of their visualization artifact and receiving automated feedback from the system. The feedback then presumably helps the designer make improvements to the artifact in the next design iteration. However, unlike existing visualization design feedback tools such as Perceptual Pat [7], which merely report metrics, Visualizationary implements the ACGT workflow from § 5.2: beyond analyzing the image, it also explains (“clarify”) the feedback using natural language, suggests ways for addressing shortcomings (“guide”), and tracks the changes to the design over multiple iterations.

This workflow is realized through five components: (1) a web-based visualization design management interface, (2) interactive design reports providing feedback at user-controlled levels of detail, (3) a revision tracker, (4) a suite of computer vision filters for extracting visualization data into symbolic form, and (5) a design guidance tool that uses an off-the-shelf Large Language Model (LLM) with a prompt preamble to explain the feedback and guide the user. The former two belong to Visualizationary’s frontend, whereas the remaining three are part of the backend.

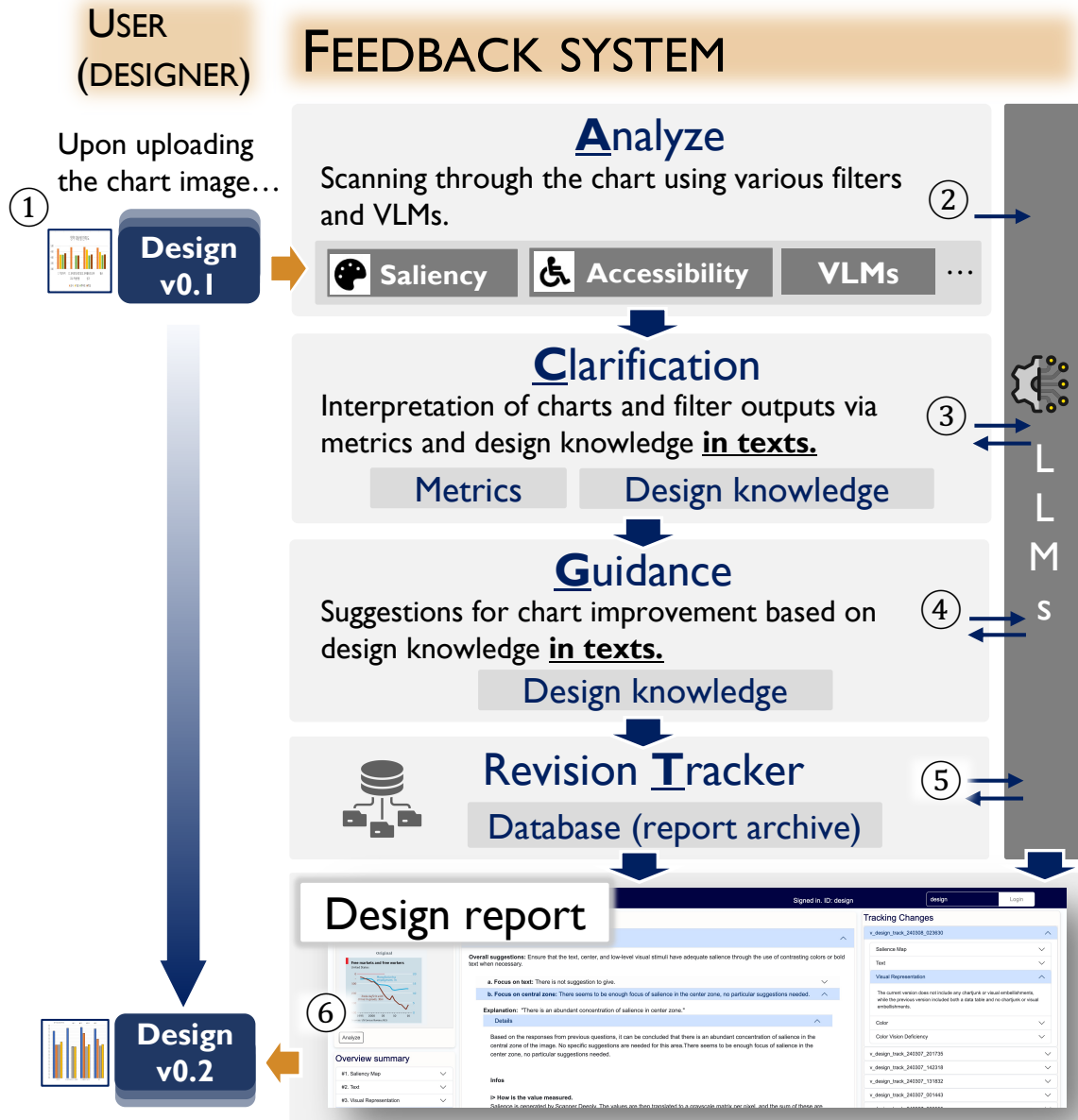


Figure 5.2: **System overview.** Visualizationary is composed of a visualization design management interface and a feedback system; the white circles refer to the analysis and report steps, respectively. The system first receives a data visualization (chart) from the project container interface ① and generates design feedback (explanations ③ and suggestions ④) based on the filters used to analyze ② the visualization. The feedback system is designed to support the *analysis-clarify-guide-track* workflow and leverages a large language model (LLM) to automate the steps. We archive the feedback generated by Visualizationary ⑤ with the revision tracker. The design report component ⑥ summarizes the feedback and the revision history for the iterative design improvement process.

Below we describe each of these components.

5.3.2 Visualization Design Management Interface

The interactive frontend for Visualizationaly is a web-based single-page application for managing the iterative design process of a visualization artifact (Fig. 5.3). A *project* is a container for a single design process, including all of the visualization revisions and their corresponding *design reports*. The frontend has operations for creating, editing, and deleting projects associated with a specific user.

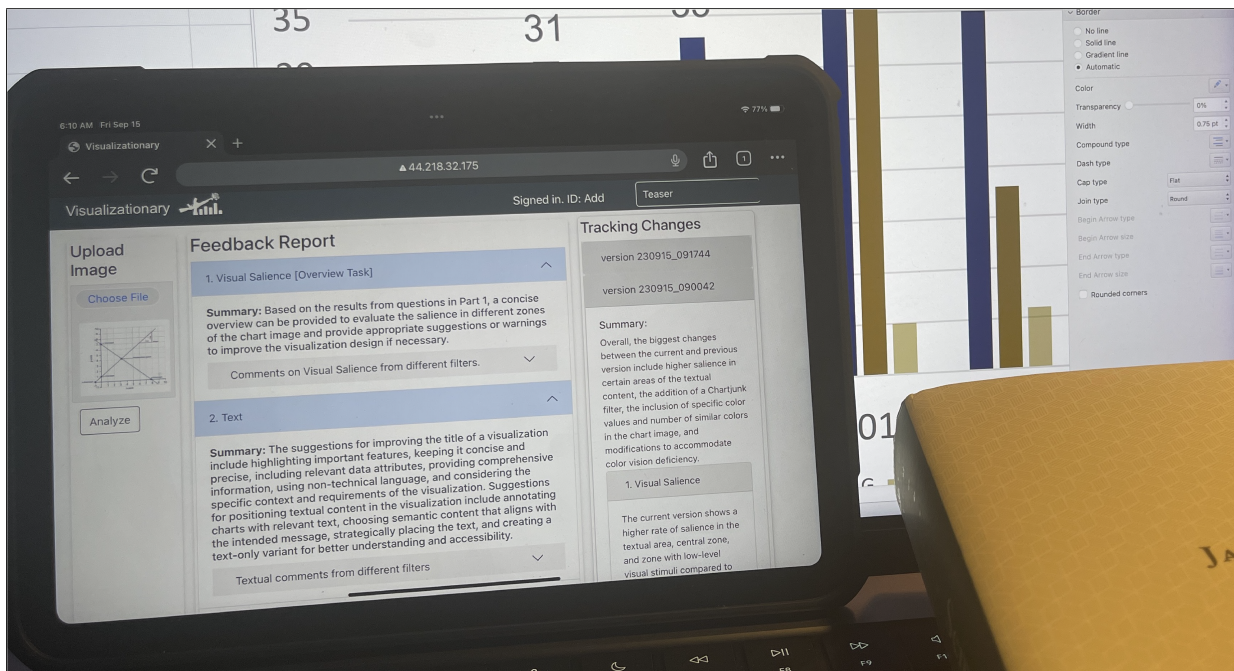


Figure 5.3: **Visualizationaly design management interface.** The system shows an overview of the timeline of revisions for a visualization artifact. Each revision consists of a screenshot of the artifact as well as an interactive design report. The right column shows the revision tracker, which provides a high-level view of the overall design trajectory.

Viewing a project shows a timeline of the revisions and allows the user to bring up the details for each. Projects can be extended by uploading a new screenshot of the visualization artifact in .jpg or .png file (see Fig. 5.1 and then launching the analyze-clarify-guide-track

workflow on the new revision. This process typically takes a couple of minutes to complete, after which a new design report is added to the timeline for the designer’s perusal. In this way, the Visualization interface is designed to manage the entire design process of a visualization artifact, from early sketches to final production.

5.3.3 Visualization Metrics

We present metrics to quantitatively assess the visualization image using visualization filters. To each evaluation category, we present the **goal**, **metric used**, and **decisions**. The decision is made based on predefined heuristics about the metric and also on the judgment of LLMs, backed by the design knowledge added as a preamble. The detailed heuristic values are described in our supplementary material. Again, note that the list is not exhaustive and can be extended.

⚙️ **Virtual Eyetracker.** We utilize the Scanner Deeply [6], a virtual eyetracker that predicts gaze on an overview task of a visualization image. We analyze the results of Scanner Deeply from three perspectives: focus of salience on text [63], center bias [96], and focus on visual attention, represented by low-level stimuli [44, 70]. The Scanner Deeply works as a tool to estimate where people would look the most when looking at a chart image. The goal is to let users foresee if the salience from the design matches their intentions.

- *Focus on text.* **Goal:** Detect visualizations that overly focus on text. **Metric used:** Use OCR to box each letter and compute the rate of salience focused in all of those boxes. We measure the amount of salience focused on text. We use OCR to detect alphabetical letters. **Decision:** Mark as “overly salient to text” if the salience in text boxes is above the 10th percentile of the chart saliency in the dataset by Shin et al. [6].

- *Focus on center.* **Goal:** Detect charts that overly focus on the center (center bias). **Metric used:** Measure the amount of salience in the center. The center zone is defined as a rectangle in the middle of the image, with its height and width as $1/3 \times$ the chart’s length and width. **Decision:** Mark as “overly salient to the center” if the center salience is above 10th percentile.
- *Focus on visual attention.* **Goal:** Detect visualizations that scarcely focus on visual attention. **Metric used:** We measure the rate of salience-covered areas of zones near (the radius is 5 pixels) where there is a difference in RGB values, which we call zones with low-level stimuli. **Decision:** Mark as “scarcely salient to the visualization” if the salience in zones with low-level stimuli is less than 90th percentile.

A Textual characters. We evaluate the existence of textual components in the visualization image [188]. Using text along with charts can help understand visualizations.

- *Title:* **Goal:** If a title is detected, then provide recommendations for the title. **Metric used:** Title is detected via DePlot, a VLM that translates visualizations into table and textual components. **Decision:** DePlot decides if there is a title in the image.
- *Textual content:* **Goal:** Detect text in visualization image. **Metric used:** Texts in the image are detected using PyTesseract-OCR. **Decision:** If the OCR tool detects at least one letter in the text, then it contains the text.

Visual representations. We observe how data is visually represented. Our focus is on providing chart recommendations based on data table and design preamble, as well as detecting *chartjunk*, or unnecessary visual embellishments [5, 189] in the design.

- *Optimal chart type:* **Goal:** Given the data, suggest an optimal chart type. **Metric used:**

The data table extracted from DePlot. **Decision:** ChatGPT makes suggestions based on a preamble of design knowledge (perceptual rankings of visual cues, described in Mackinlay's APT [108]), and the data table provided by DePlot. To help the designer make an optimal decision, we provide the table as well as the design knowledge instilled in ChatGPT.

- *Visual embellishment* (“chart junk”): **Goal:** Detect chartjunk in the visualization image. **Metric used:** A computer vision model that detects objects, such as YoloR [154]. **Decision:** The object detection model detects objects within the chart image.

🎨 **Color perception.** Detect colors and warn of misusing colors when multiple colors or similar colors are used [101, 121].

- *Variability of colors:* **Goal:** Identify all colors used in the visualization image. **Metric used:** A customized function created in Python. **Decision:** The function identifies multiple colors.
- *Similarity of colors:* **Goal:** Identify similar colors in the visualization image. **Metric used:** A customized function created in Python. A group of similar colors are determined using differences in RGB values between detected colors. **Decision:** The function identifies a group of similar colors.

♿ **Accessibility.** It is also important to create visualizations that are accessible for everyone [149, 172].

- *Color vision deficiency:* **Goal:** Detect visualization images that are hugely affected when simulating the image into how people with color vision deficiency (deuteranopia, protanopia, tritanopia) would see. **Metric used:** We measure the information loss caused

in simulated visualization images, using the difference in entropy loss from its original visualization image. **Decision:** If the entropy loss is larger than a threshold in simulated visualization images, then, warn of the loss in information for people with color vision deficiency.

We instill these metrics along with the suggestions relevant to the decisions into ChatGPT as a prompt (see § 5.3.6). Then, based on a series of template questions, ChatGPT provides the appropriate answer.

5.3.4 Visualization Design Reports

Results from the feedback backend are presented as interactive *visualization design reports*. A design report is an interactive document consisting of a hierarchy of expandable sections designed in a bottom-up fashion. Sections are organized based on the main categories of visualization metrics in § 5.3.3. The contents of a section in the report include both natural language and illustrative images (such as heatmaps, extracted shapes, or gaze maps) (see Fig. 5.1 (C)). Each section provides a short overall summary of suggestions (Fig. 5.1 (C) ②) of the underlying data for that metric followed by two standardized paragraphs:

- **Explanations:** Explanation of the visualization feedback in natural language (“clarify” in the ACGT workflow) (e.g., Fig. 5.1 (C) ④).
- **Suggestions:** Recommendations addressing shortcomings in the visualization for that specific metric (“guide”) (e.g., Fig. 5.1 (C) ③).

Expanding the subheadings yields additional explanations (see Fig. 5.1 (C) ⑤) and suggestions. Sections can be expanded down to raw visual metrics (see Fig. 5.1 (C) ⑥). We also provide

additional information and warnings about each filter, to help the designer make decisions while reading the report. The rationale for this design is to present feedback in a summarized form that can be easily navigated. It also allows experienced designers to see high-level themes without having to drill down into low-level details.

We provide additional functionality to help navigating the report. We indicate sections that require modification using a circular visual mark, either yellow or red next to the title of each section (see Fig. 5.1 (C) ①). If there exist at most one issue in a section, then the corresponding circle is yellow. If there is more than one, then the circle is orange. We provide a short overview summary of responses per category in Fig. 5.1 (E). This aids designers in quickly identifying their topics of interest by skimming section summaries before delving into specifics.

5.3.5 Revision Tracker

The *revision tracker* gives a high-level overview of changes from one revision to another for the entire design process. This component realizes the “track” in the ACGT workflow. The intention is to give the designer a bird’s eye view of the design process, thereby seeing the improvements over time and also avoiding local optima. The natural language in the tracker is derived using the LLM with the reports to be compared as prompts; see below.

5.3.6 Prompt Engineering for Visualization Feedback

We use off-the-shelf LLMs, such as ChatGPT, to generate the interpretations and suggestions in the visualization design reports. However, formulating effective prompts requires careful *prompt engineering* [176, 190–193]. Prompt engineering is typically based on empirical data spe-

cific to each model, and our approach using LLMs for visualization feedback is no different. An LLM is not deterministic, and may occasionally produce unexpected output. It also tends to be quite verbose, producing long responses to even short and well-contained queries. We thus refer to guidelines from prior work [194] in designing prompts to avoid such non-deterministic and long-winded behavior:

- *Use exact keywords*: concepts as reserved words in programming;
- *No synonyms*: many labels for same concept yields nondeterminism;
- *Avoid unnecessary information*: irrelevant data expands the scope;
- *Constrain feedback length*: avoid rambling and unfocused answers;
- *Ask specific queries*: limit the scope of the query; and
- *Avoid open-ended queries*: encourage short and direct responses.

We construct two types of prompts: *analyze-clarify-guide* prompts and *track* prompts. To create those prompts, we define the following templates T (the entire list of prompts is included in OSF³):

³<https://osf.io/v7hu8>

$$\begin{aligned}
T_{acg} &= [\text{Q}] + \text{"Solve the problem based on this guideline:"} + \\
&\quad [\text{cond}] + [\text{filter-suggestions}]. \\
T_i &= [\text{Q}] + \text{"Here are details about the current version:"} + \\
&\quad [\text{curr-output}] + [\text{curr-interpretations}] \\
&\quad + \text{"Here are details about the previous version:"} + \\
&\quad [\text{prev-output}] + [\text{prev-interpretations}].
\end{aligned}$$

ACG-template. We call the analyze-clarify-guide (ACG) prompt template T_{acg} . $[\text{Q}]$ is the question we define to create interpretations and suggestions. For instance, we use “*Analyze the visual salience on text. Provide interpretations in 2 sentences.*” or “*Provide suggestions about the result of the previous question in 2 sentences.*” We define 9–12 questions for each visualization metric. We replace the metric name in different questions, e.g., from visualization salience to color blindness. We call the prompt conditions $[\text{cond}]$: “*Please interpret exactly in the following way, as if you are an assistant to a visualization designer, explaining to novice visualization designers. If no visual salience is detected, then just interpret it as No salience is detected in the chart image. If (the rate of salience in the textual zone over the rate of the textual zone in chart image) from the measured result is less than 0.6, then interpret it as a lack of salience in textual elements.*” At the end of each prompt, we append $[\text{filter-suggestions}]$, a compact representation of visualization design knowledge in natural language. $[\text{filter-suggestions}]$ is available in the supplementary material.

T-template. T_t is used for generating track (T) prompts. We define the question [Q] for extracting differences between two different versions of data visualization as follows: “*Given the information, in one sentence, concisely, what are the changes made between the current and previous versions about visual salience? Although the changes may seem minor, try to describe even the small, acute changes made between the current and previous versions, in terms of the visual salience.*” `[{curr/prev}-output]` is the collection of visualization design knowledge, expressed as natural language, calculated by the visualization filter components. `[{curr/prev}-interpretations]` is the interpretations Visualizationary generated for the two visualizations.

5.3.7 Implementation Notes

The Visualizationary tool is implemented as a client-server web application using HTML, CSS, JavaScript, and JQuery. We use the Python Flask web framework⁴; the analysis components were implemented server-side in Python 3. We store data into MongoDB.⁵ During the user study, Visualizationary was hosted on Amazon Web Services (AWS). We choose ChatGPT using `gpt-3.5-turbo` as the LLM.

5.4 User Study

We performed a user study to assess the impact of LLM-generated feedback in the visualization design pipeline using the Visualizationary tool. We opted for a qualitative study where designers of different skill were asked to use Visualizationary to support the design process of a

⁴<https://flask.palletsprojects.com/>

⁵<https://www.mongodb.com/>

new visualization over several days. We then let a small group of visualization experts (with more than 10 years of experience in the field) assess the design process without seeing the Visualizationary reports. We also interviewed our participants to collect their perception of the impact of Visualizationary on their design work. The study received approval from our university's ethics review board (IRB).

Table 5.1: **User study demographics.** A total of 13 visualization designers of different seniority participated in the user study. These participants were all above the age of 18, had at least a bachelor’s degree in visualization or related field, and had more than three years of experience in designing visualizations. We categorize those with less than or equal to 3 years of experience in visualization as novice (6 participants), those with 4 to 7 years as intermediate (4 participants), and those with more than 7 years as expert groups (3 participants), respectively.

ID	Gender	Age	Education	Job Title	Years in Vis	Group
P1	Male	35	Ph.D.	Postdoc in Visualization	11 years	Expert
P2	Male	26	Bachelor’s Degree.	Ph.D. student in Computer Science	3 years	Novice
P3	Male	34	Ph.D.	Research scientist	5 years	Intermediate
P4	Male	29	Master’s Degree	Software engineer	3 years	Novice
P5	Male	28	Master’s Degree	Ph.D. student in Computer Science	6 years	Intermediate
P6	Male	31	Master’s Degree	Ph.D. student in Computer Science	6 years	Intermediate
P7	Male	30	Bachelor’s Degree	Ph.D. student in Computer Science	5 years	Intermediate
P8	Male	33	Ph.D.	Assistant professor in Visualization	10 years	Expert
P9	Male	31	Master’s Degree	Data scientist	9 years	Expert
P10	Female	24	Bachelor’s Degree	Master’s student in Data Science	3 years	Novice
P11	Male	33	Bachelor’s Degree	Software engineer	1 year	Novice
P12	Female	33	Bachelor’s Degree	Primary school teacher	1 year	Novice
P13	Male	32	Ph.D.	Postdoc in Computer Science	1 year	Novice

5.4.1 Participants

We advertised our study through messages to the Data Visualization Society, to alumni of our institution, and to other academic institutions. Out of many applicants, we selected only those who are over the age of 18, are capable of reading and speaking English fluently, and have experience in visualization for more than three years. As a result, we were able to recruit 13 visualization designers of different seniority. Table 5.1 shows the demographics of the participants. These participants were paid a \$40 Amazon gift card upon concluding the study.

We note that while Visualizationary was originally designed for novice and intermediate visualization designers, our list of participants also included expert ones. We included such participants because we were interested in seeing whether our tool could benefit even an expert. Furthermore, we also wanted to eliminate the confound of inexperienced designers having to first learn how to author visualizations.

Beyond the designer participants, we also recruited three expert participants with at least seven years of experience in visualization: an associate professor (or equivalent), an assistant professor, and a Ph.D. candidate (within months of graduation) within the research field. These experts only took part in the final independent design assessment step.

5.4.2 Task

Our study asked participants to design a new visualization over a period of 3–5 days while using the Visualizationary interface to support their design process. Participants were free to choose a dataset and to craft any type of visualization using any tool or combination of tools. We asked each participant to upload at least 5 versions (including the first and last versions) of their

designs into Visualizationary. We imposed the following requirements on participants:

- They must spend at least two hours on the design process;
- They should work independently on the design task;
- Images must be between 100×100 and 2000×2000 pixels;
- Images were uploaded in either .png or .jpg formats;
- No photographs allowed; and
- No confidential information or data was used.

We also secured permission to use the designs for this publication.

5.4.3 Procedure

The study was performed exclusively on the Internet using video conferencing and consisted of four steps. We obtained participant demographic information and scheduled the dates for each part during the recruitment process. Below we describe each of the steps in detail.

Step 1: Pre-study Briefing. We started each session with an overview of our study, answered any questions from the participant, and collected informed consent using an online consent form, as per the directives of our IRB. Then the experiment administrator explained the longitudinal chart design task (§ 5.4.2). Next, the administrator showed how to operate the Visualizationary system: creating an account, logging in, uploading screenshots, and accessing design reports. The administrator answered any additional questions from the participant. The study was concluded by finding an appointment for the post-interview. On average, this briefing process took 30 minutes.

Step 2: Longitudinal Chart Design. Participants were then given at least three days to design their visualization, during which time they were asked to upload at least five versions of their visualization. We imposed no restrictions on how participants allocated their time but asked that they dedicate at least a total of two hours during the period on the design and that they use the Visualizationary interface. During this period, the experiment administrator was available to answer any technical questions the participants had.

Step 3: Post-study Interview. We conducted a post-interview interview with each participant at the end of the design period. We started by asking each participant to explain how their designs evolved over time. We always selected the first and last version of the visualization artifact. If more than five versions were uploaded by a participant, we asked the participant to identify the three most significant versions other than the first and last ones. We defined “significant” as versions where feedback had caused them to make radical changes to the design. Following that, we posed questions about their experiences using Visualizationary during the design process. Afterward, we asked about their concepts of an ideal feedback provider after the study. The questions we asked are shown in the next section. While we allocated 30 minutes for this session, some of the participants exceeded this time limit.

Step 4: Expert Design Assessment. In the final step, our independent expert participants (all with experience teaching data visualization and grading visualization assignments) were given all five designs for each of the designer participants. They were asked to provide their assessment of the quality of changes during the design process from the first to the final design for each participant using a 1–5 Likert scale (1 = significant decline in quality, 3 = neutral, 5 = significant

quality improvement). They were also asked to motivate their assessment using 1 to 2 sentences. This meant that each expert gave 1 Likert scale rating and 1-2 assessment sentences per designer participant.

5.4.4 Data Collection

Interviews were video and audio recorded. We transcribed audio for further analysis and interpretation. We collected demographics using online forms before the study. All uploaded chart images and design reports were collected automatically by the Visualizationary interface.

5.4.5 Apparatus

The interview and the experiment were conducted using participants' own personal computers. Although we did not enforce strict hardware constraints, we recommended participants use computers with a resolution of at least 1920×1080 and use Google Chrome, as the Visualizationary interface is optimized for that setting. When designing visualizations, the participants had the freedom to choose their favorite authoring tools, as long as they could upload a screenshot of their visualization image in .jpeg or .png format.

5.5 Results

We now present the results from our longitudinal study, including design iterations, subjective comments, and observations.

5.5.1 Design Evolution

Figs. 5.4, 5.5, 5.6, and 5.7 present visualization snapshots from three representative participants: novice, intermediate, and expert groups. We only show three examples due to page limits; the complete snapshots and descriptions from all participants can be found in our supplementary material. We asked participants to explain the changes created in each iterative step during the interview and describe their intent in making changes below. Below we discuss the design evolution for each participant as follows: the visualization, authoring tool, and dataset used, a summary of each version, including the feedback from Visualizationary, the changes made, and—if given—their rationale.

Participant 1. P1 created a boxplot using Microsoft Excel to compare the performance of their own learning model with other existing models using a boxplot. They used the Excel chart gallery to create the initial version.

- **Version 1→2:** *Feedback:* Visualizationary reported that there was no text or labels in the chart and that the salience map focuses only towards the center, not on other parts of the chart. *Revision:* P1 added a title and legend, widened the chart, and applied categorical colors.
- **Version 2→3:** *Feedback:* Visualizationary noted that too many colors were used, and suggested that colors be kept minimalistic. *Revision:* P1 changed the color of boxplots to a gray scale, placed texts below the horizontal axis after removing the legend, and highlighted the rightmost text in red.
- **Version 3→4:** *Feedback:* Visualizationary noted that many similar colors were used, and that the salience map still focuses towards the center overly. Visualizationary suggested attracting

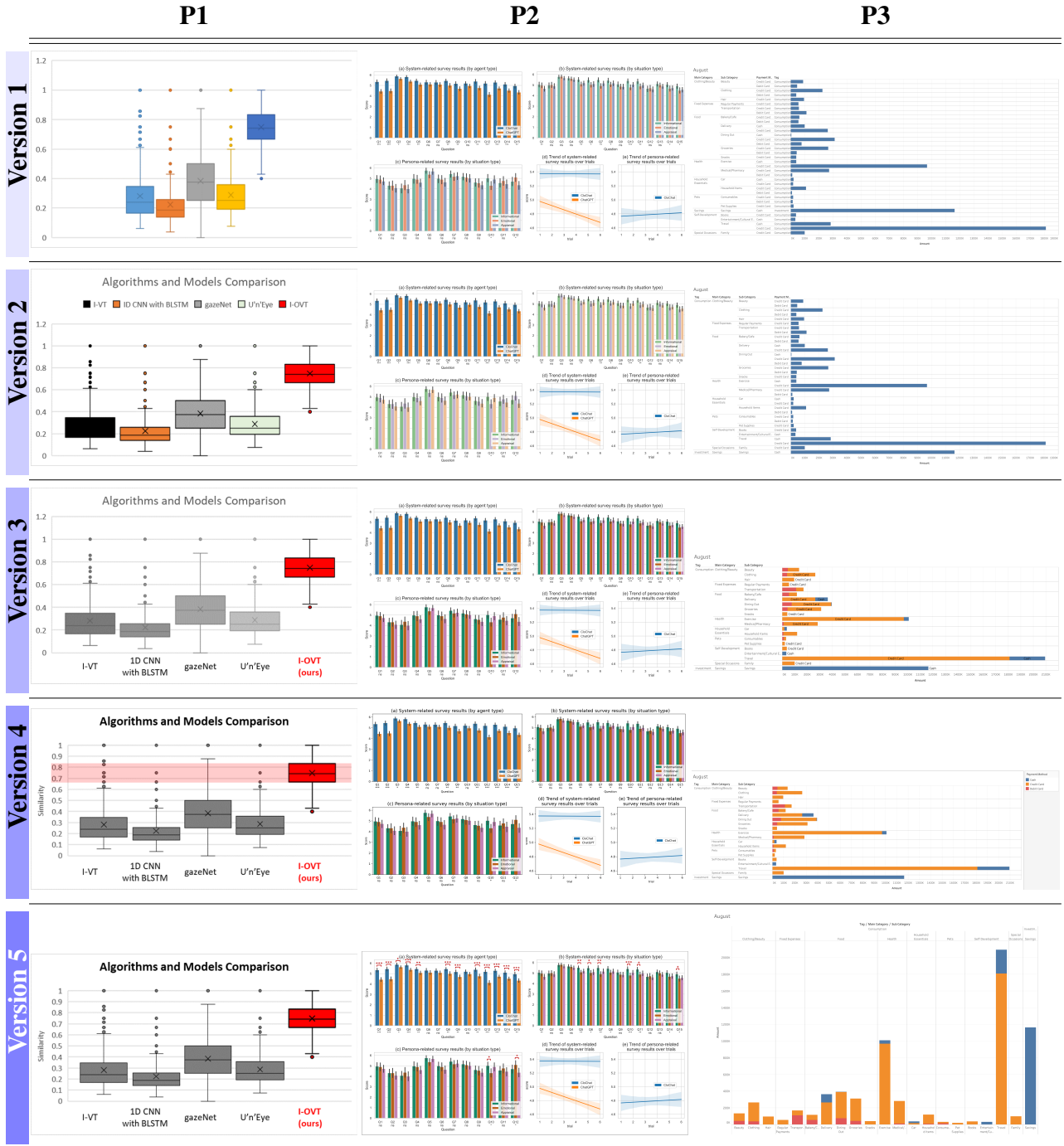


Figure 5.4: **Design evolution (P1-P3).** Five versions (as identified by the participants) during the design process of creating a new visualization artifact from scratch using Perceptual Pat. The first three participants from the study are represented.

the viewer's attention to a point of interest by highlighting. *Revision:* P1 adjusted all boxplot colors to gray, emphasized a specific area, and bolded the title and the red boxplot.

- **Version 4→5: Feedback:** The Visualization saliency map seemed to overly focus on the

red-highlighted zone. *Revision:* P1 changed the colors of boxplots to a uniform gray color and removed the highlighted zone. *Rationale:* P1 did not want viewers to focus overly on the red-highlighted area, but look into other parts of the chart, too.

Participant 2. P2 generated a collection of 5 charts using Matplotlib, a Python library to visualize empirical data, for presentation in a journal paper.

- **Version 1→2:** *Feedback:* Visualizationary classified colors of two bars (blue and green) as similar, and that there should be a title that represents all charts. *Revision:* P2 changed colors using a new color palette, but did not add a title. *Rationale:* P2 did not feel the need to add another title.
- **Version 2→3:** *Feedback:* Visualizationary showed that the updated colors had color-vision deficiency issues (protanopia). *Revision:* P2 changed the colors of two bar charts using a palette suggested by Matplotlib.
- **Version 3→4:** *Feedback:* The saliency map showed too much focus on the center of the charts, not being salient in intended parts of the chart. *Revision:* P2 removed the grid to see if the saliency focuses more on the data representation.
- **Version 4→5:** *Feedback:* Visualizationary reported that the chart lacks textual content. *Revision:* P2 added stars on top of the bars that that needed to be emphasized, and restored the grid lines. *Rationale:* P2 thought that eliminating grids made it difficult to compare between values.

Participant 3. P3 wanted to analyze their monthly spending based on two factors: (1) category, and (2) payment method. They used Tableau to author the visualization.

- **Version 1→2:** *Feedback:* The salience map in Visualizationary asserted that the focus on text is higher than other charts. *Revision:* P3 changed the order of the dataset hierarchy to remove repetitive keywords.
- **Version 2→3:** *Feedback:* Again, Visualizationary stated that there is too much focus on text. *Revision:* P3 removed the “payment method” category, consolidated its values into a stacked bar chart, and labeled each bar segment with its corresponding payment method.
- **Version 3→4:** *Feedback:* And once again, Visualizationary said that there is still too much focus on text. *Revision:* P3 eliminated the text alongside the bar chart and introduced a legend instead.
- **Version 4→5:** *Feedback:* And again, Visualizationary maintained that there is still too much text. *Revision:* P3 rotated the bars vertically and separated the labels for categories and sub-categories.

Participant 4. P4 used Matplotlib to draw the distribution of certain keywords in three different datasets: A, B, and C.

- **Version 1→2:** *Feedback:* Visualizationary indicated low color contrast, and that using different colors could attract attention more effectively. *Revision:* P4 used three colors (blue, red, and green) to represent different datasets.
- **Version 2→3:** *Feedback:* Visualizationary identified that there was no title, and that the salience map indicated overly focus at the center when there is not much to gain from

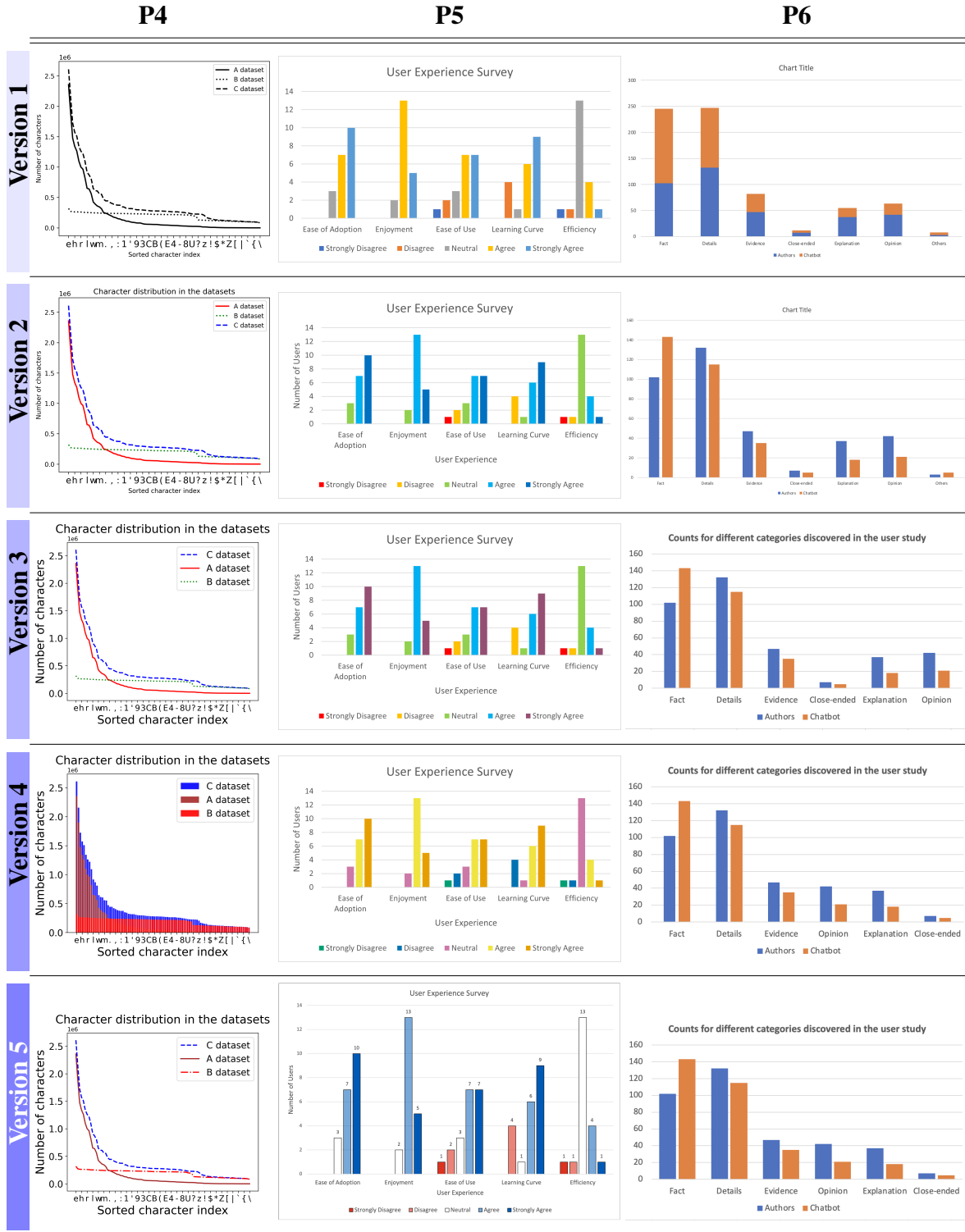


Figure 5.5: **Design evolution (P4-P6).** Five versions (as identified by the participants) during the design process of creating a new visualization artifact from scratch using Perceptual Pat. The next three participants from the study are represented.

viewing the center. *Revision:* P4 added a chart title. *Rationale:* P4 ignored the center bias issue.

- **Version 3→4:** *Feedback:* Visualizationary recommended using larger fonts to add clarity, and also suggested a bar chart based on the type of the dataset. *Revision:* P4 enlarged fonts of texts in the chart, and changed into a stacked bar chart. *Rationale:* Initially, P4 had not considered bar charts, but found them fitting for nominal data.
- **Version 4→5:** *Feedback:* Visualizationary commented that people with deuteranopia may not clearly distinguish colors. *Revision:* P4 returned again to the line chart, and changed colors of A, B datasets. *Rationale:* P4 thought that the line chart was easier to compare the distributions than the stacked bar chart.

Participant 5. P5 used Microsoft Excel to visualize a dataset collected to evaluate user experience.

- **Version 1→2:** *Feedback:* Visualizationary warned that the color in some parts were similar, and suggested that texts should be added to facilitate the understanding of charts. *Revision:* After identifying that the colors ‘strongly disagree’ and ‘strongly agree’ were similar, they changed palettes. Also, they added label in each axis.
- **Version 2→3:** *Feedback:* Visualizationary still warned of using similar colors within the visualization. *Revision:* P5 shifted the ‘strongly agree’ color from blue to purple for clearer contrast with ‘agree’.
- **Version 3→4:** *Feedback:* Visualizationary warned of information loss for people with color vision deficiency. *Revision:* P5 changed colors using color vision deficiency-friendly colors

using palettes from the web.

- **Version 4→5:** *Feedback:* The tool still cautioned of the issues on color vision deficiency, and suggested to add more textual explanations, and informed of the over-abundant focus at the center. *Revision:* P5 modified the bar colors to a neutral for white, while using varying saturations of blue for agreement and red for disagreement. The blue-red combination was a friendly color for color vision deficiency. P5 added texts on top of each bar, and expanded the chart vertically to diverge attention.

Participant 6. P6 employed Microsoft Excel to contrast human and conversational AI efficiencies, starting with a recommended stacked bar chart.

- **Version 1→2:** *Feedback:* The tool suggested bar chart as the preferable chart based on the data and Mackinlay's work [195]. *Revision:* P6 changed from stacked bar chart to a bar chart.
- **Version 2→3:** *Feedback:* Visualizationary was not able to detect title, suggested to reduce the impact of text in charts, but to increase the font size to add clarity. *Revision:* P6 added title, and increased font size. *Rationale:* P6 thought that text was wrongly detected, so did not think that there needed to be more text.
- **Version 3→4:** *Feedback:* Visualizationary noted that there was an abundance of salience at the center in the salience map, and that the chart used too many colors. *Revision:* P6 changed the order of features in the horizontal axes. *Rationale:* P6 did not believe that they were using too much colors.
- **Version 4→5:** *Feedback:* Visualizationary said that the visual components were less em-

phasized than the text. *Revision:* P6 increased the width of all bars for improved visual representation.

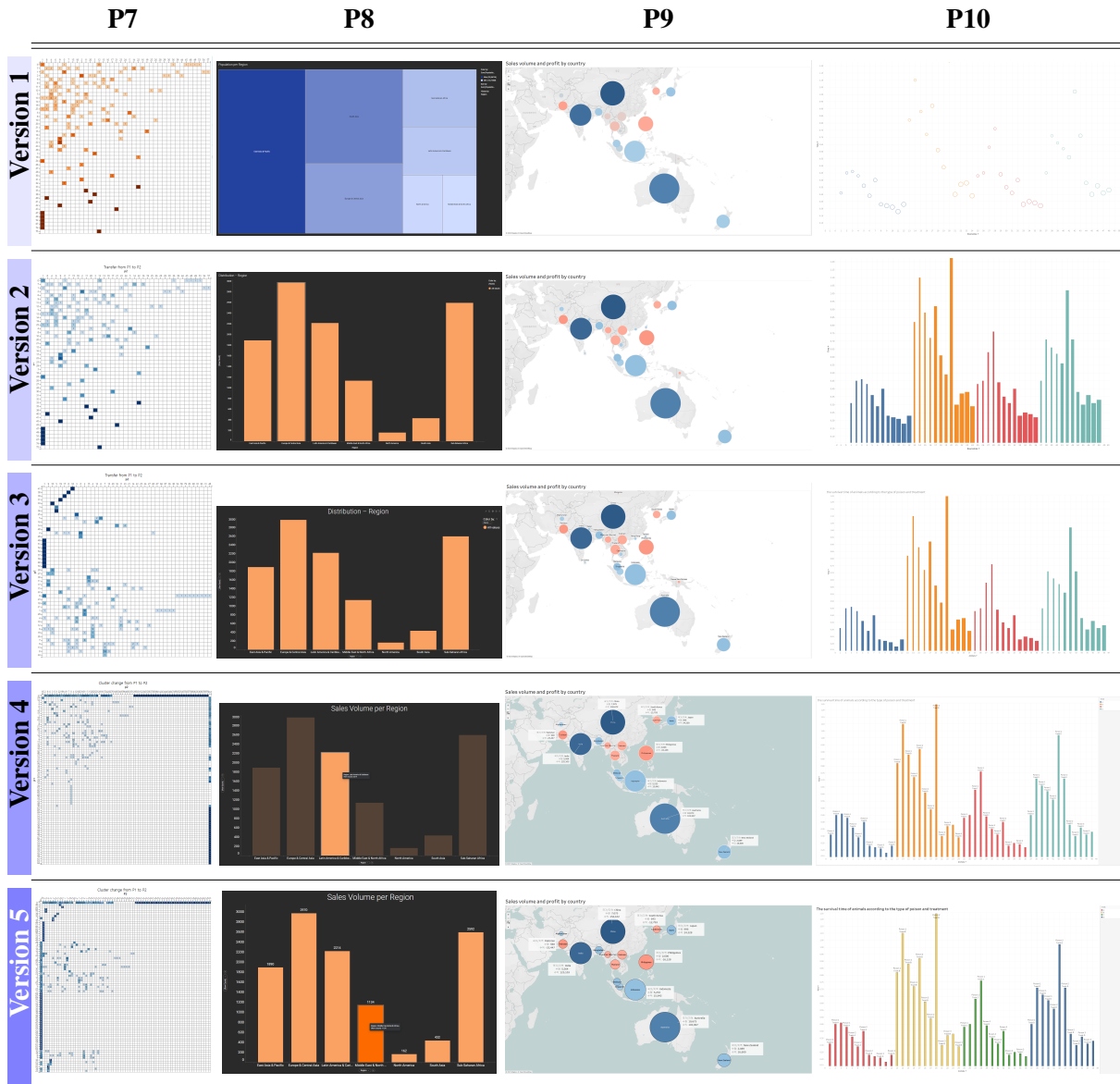


Figure 5.6: **Design evolution (P7-P10).** Five versions (as identified by the participants) during the design process of creating a new visualization artifact from scratch using Perceptual Pat. The last four participants from the study are represented.

Participant 7. P7 used `D3.js` to create a visualization for analyzing character distributions using a heatmap.

- **Version 1→2:** *Feedback:* Visualizationary was not able to detect the title, and mentioned that the salience is focused too much towards the center, but P7 wanted to identify anomalous parts. *Revision:* P7 added the title, and changed the color from orange to blue.
- **Version 2→3:** *Feedback:* From the tool's perspective, the saliency map still was overly focusing towards the center. The tool also suggested bar chart as the desirable choice for chart. *Revision:* P7 used hierarchical clustering to group zones with high frequency values together. *Rationale:* P7 was not convinced to use bar charts instead of a heatmap.
- **Version 3→4:** *Feedback:* The saliency map focused towards the center. *Revision:* P7 increased the ranges of horizontal and vertical axes in the heatmap. *Rationale:* P7 wanted the saliency map to focus on dark-colored zones.
- **Version 4→5:** *Feedback:* The saliency map retained its center-bias. *Revision:* P7 modified the heatmap's distribution for more even distribution.

Participant 8. Using Spotfire, P8 analyzed the coffee sales revenue by continent using a treemap.

- **Version 1→2:** *Feedback:* Visualizationary recommended bar chart to present the data. *Revision:* P8 changed the data to a bar chart. *Rationale:* P8 agreed that the bar chart was more effective than a heatmap for comparison.
- **Version 2→3:** *Feedback:* The tool was not able to detect the title. *Revision:* P8 moved the title to the center and enlarged its font size. *Rationale:* The undetectable title prompted P8 to emphasize chart text for clarity.
- **Version 3→4:** *Feedback:* Visualizationary proposed adding highlights to a region of interest, suggested effective titles, and also recommended maintaining minimalistic perspective

in describing charts. *Revision:* P8 darkened the chart except a region of interest, refined its title for data representation, and removed the legend.

- **Version 4→5:** *Feedback:* The salience shown in Visualizationary focused too much towards the center of the chart, where the area of interest is located, and suggested to use text to make the chart more understandable. *Revision:* P8 reduced highlighting by using dark orange for the area of interest, and added values on top of each bar. *Rationale:* P8 believed the area of interest was overly emphasized, and did not want people would not look into other parts of the chart.

Participant 9. Using Tableau, P9 chose to display a multinational company's profits in Asia on a geographic map.

- **Version 1→2:** *Feedback:* From the protanopia-simulated image in the tool, several circle cues were hard to detect. *Revision:* The colors in each circle were made darker and clearer.
- **Version 2→3:** *Feedback:* The salience map in Visualizationary directed attention country names, but several were unrelated to the data. *Revision:* Country names were added on top of each circle, and irrelevant country names originally on the map were removed.
- **Version 3→4:** *Feedback:* Visualizationary noted that textual content would make the chart more readable. *Revision:* P9 relocated country names to circle centers, added explanatory boxes, and differentiated ocean colors for clarity.
- **Version 4→5:** *Feedback:* Still, the images simulated for color vision deficiency showed that some circles were not clear. *Revision:* Strokes were added to circles.

Participant 10. P10 employed Tableau to study the impacts of four medicine concentrations on animal lifespans using a bubble chart.

- **Version 2:** *Feedback:* The tool recommended using bar chart. *Revision:* P10 changed chart type into a bar chart.
- **Version 3:** *Feedback:* The tool suggested using magnification to highlight the difference between low and high values, and noted that no title was detectable. *Revision:* P10 magnified the vertical axis and added a title.
- **Version 4:** *Feedback:* Perceptual Pat asserted that textual information helped understand charts. *Revision:* P10 added labels on top of the visualization.
- **Final version:** *Feedback:* Perceptual Pat stated that rainbow colors worked well for presenting discrete information, and also suggested to enlarge the font to clarify the textual content. *Revision:* P10 changed the color of bars using rainbow palette, and made the text in bold.

Participant 11. P11 created a stacked area chart (recommended by Microsoft Excel) to show time spent on daily activities.

- **Version 1→2:** *Feedback:* Visualizationary recommended bar charts to facilitate the comparison of values. *Revision:* P11 changed from stacked line chart to a stacked bar chart.
- **Version 2→3:** *Feedback:* Visualizationary noted that there is too much focus of salience towards the center, and recommended that the attention be diverted to other places of the designer's interest, if needed. *Revision:* P11 adjusted the chart's size ratio and incorporated

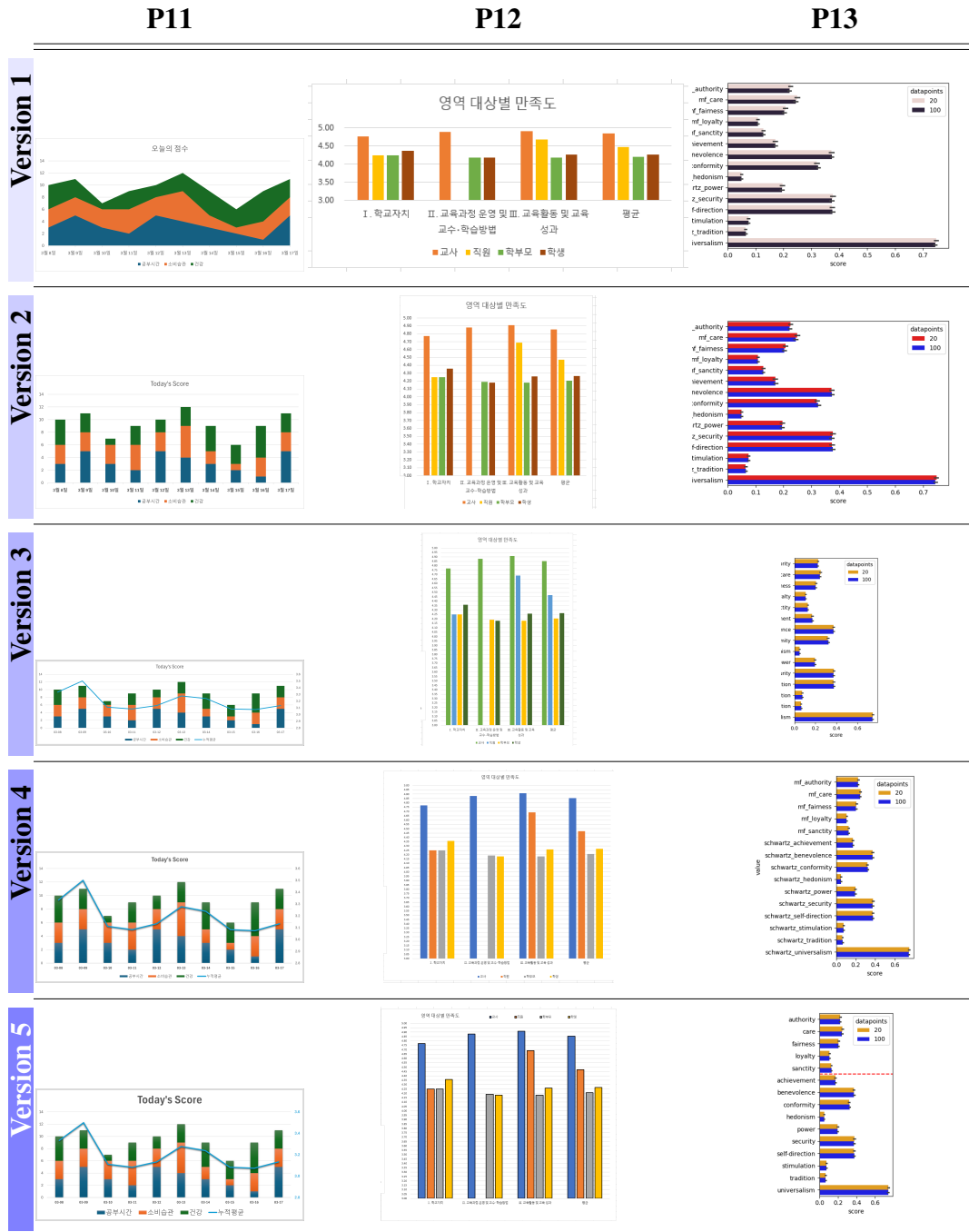


Figure 5.7: **Design evolution (P11-P13)**. Five versions (as identified by the participants) during the design process of creating a new visualization artifact from scratch using Visualizationary. The next three participants from the study are represented.

a new line chart, which displayed the cumulative average hours spent on a specific activity (“work”). P11 introduced a vertical axis to represent the line value in the chart. *Rationale:* For adding the line chart, P11 wanted to examine influences on work.

- **Version 3→4:** *Feedback:* Again, Visualizationary observed excessive concentration of salience in the central area. *Revision:* P11 re-adjusted the chart dimensions, and highlighted legend and title to divert the attention.
- **Version 4→5:** *Feedback:* Saliency is overly focused on the center. *Revision:* P11 increased title and legend font sizes.

Participant 12. P12 also used Microsoft Excel to create a bar chart from a multidimensional data table.

- **Version 1→2:** *Feedback:* Visualizationary alerted the overabundance of saliency in the center. *Revision:* In order to divert the attention towards the center, P12 enlarged the size, and changed the size ratio of the chart.
- **Version 2→3:** *Feedback:* Visualizationary noted that there is information loss in the color vision deficiency filter, and that there was still abundance of salience in the center of the chart image. *Revision:* P12 enlarged the size of the chart vertically, and changed the color of chart image.
- **Version 3→4:** *Feedback:* According to Visualizationary, there was still noticeable amount of loss in color vision deficiency, and the abundance of salience in the center has not been solved. *Revision:* P12 changed the color of the chart, enlarged vertically, and increased the size of texts and ticks.
- **Version 4→5:** *Feedback:* Again, Visualizationary says that the salience is overly focused towards the center, and that to highlight other parts of the chart that needs attention. *Revision:* P12 added strokes to bars.

Participant 13. P13 used Matplotlib library in Python to analyze the results from conducting sentimental analysis about Twitter data of a particular day and region.

- **Version 1→2:** *Feedback:* Visualizationary noted that attention is centered in the visual and recommends employing contrasting colors to shift focus. *Revision:* P13 changed the color to red and blue to add more contrast to the visualization.
- **Version 2→3:** *Feedback:* Visualizationary observed excessive central focus and highlighted potential data loss for color vision-deficient viewers. *Revision:* P13 changed the size ratio of the chart, and changed the color from blue-red to blue-orange.
- **Version 3→4:** *Feedback:* Visualizationary said that the salience map is not focusing much textual information. *Revision:* P13 readjusted the size of the chart, and added labels for axes.
- **Version 4→5:** *Feedback:* Visualizationary mentioned that now there is too much salience in text. *Revision:* P13 added a red line, and shortened labels. *Rationale:* P13 introduced a red line to the chart to facilitate his personalized analysis.

Table 5.2: **Questions for post-study interviews.** We asked participants a total of 7 questions about their experience in using Visualizationary to understand the role of Visualizationary in the design process, to unveil their decision-making process in acting on feedback, and to evaluate the tool.

QUESTION TYPE	SUBJECT	QUESTIONS
Role	Analyze	How was your chart problem detection aided by Visualizationary?
	Clarify	How did Visualizationary assist in clarifying chart guidance or analysis?
	Guide	How was the guidance from Visualizationary for visualization design?
	Track	How did tracking influence your design modifications?
Decision	Judgment	In general, when do you act on feedback or decline it?
Evaluation	System	What are the pros/cons of Visualizationary when the LLM bases answers on established visualization design knowledge?
	Impression	Overall, how did you like the tool?

Table 5.3: **Evaluation of design evolution.** We asked three external experts to evaluate the improvements in the designs generated by 13 participants. Scores are measured using a 1–5 Likert scale, where 1 represents a significant decline, 3 is neutral, and 5 is a significant improvement. Each value shows the mean value of scores from external evaluators; values in parentheses are standard deviations.

Part.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	Avg
Avg.	4.67	2.67	3.67	3.00	5.00	4.33	2.67	3.00	4.67	4.00	3.67	2.67	4.00	3.69
S.D.	(0.58)	(0.58)	(0.58)	(0.00)	(0.00)	(0.58)	(0.58)	(1.73)	(0.58)	(0.00)	(0.58)	(0.58)	(0.00)	(0.83)

5.5.2 Post-Study Interview

After the experiment, we conducted a post-interview session to examine closely the iterative improvements of the design and the role of Visualizationary in making the changes. Our initial set of questions (Table 5.2) aimed to reacquaint the participants with the changes their designs underwent and to understand the underlying reasons behind them. For every design refinement, we asked participants to elaborate on the following: (1) the **changes** in the design, (2) the **reasons** for the changes in the design, and (3) the **role of Visualizationary** in the changes. We then transitioned to understand their experiences in using Visualizationary during the design process, and their perspective on design feedback mechanisms. Our goals were the following:

1. **Role.** Understand Visualizationary’s role in every ACGT step.
2. **Decision.** Identify patterns when designers accept/reject feedback.
3. **Evaluation.** Evaluate the strengths and weaknesses of the tool.

Detailed below are the questions presented to our participants.

Role for Analyze: Expert Analyst. Visualizationary is capable of not only detecting issues, but also in providing relevant knowledge that only experts can have. Some of the observed comments were aspects participants had not initially considered, yet were contributing to the improvement of the visualization. In other instances, the tool’s feedback was on aspects participants were famil-

iar with or had previously encountered, but which were typically overlooked during visualization design, such as the color deficiency filters. P2 said, “[the tool] is capable of showing problems that I was not aware of, although I should have been.” Upon diagnosis, the tool supplies relevant data from visualization design knowledge, which, as P8 noted, is information that even experts might occasionally miss. Said P8, “even experts may not be aware of those specific details about visualization principles.” However, there were also times when the tool misdiagnosed the chart image, resulting in potentially out-of-context feedback.

Role for Clarify: Organizer. Because the output of this report is generated by ChatGPT, it is predominantly textual. The hierarchical layout of the design allowed participants to view the summary of each section, selectively pick the issues that they found relevant and made sense, and drill down to the details they were interested in. 10 out of 13 participants felt that this capability helped them be more organized.

Role for Guide: Guidepost. Given a detectable issue, Visualizationary is capable of providing general suggestions on how to address them, but does not always deliver specific actionable feedback. This means that human intervention is still required to find a solution. Participants also noted variability in the guidance: some feedback was easy to implement whereas other feedback required a substantial amount of effort to address. Easily addressed were tasks that can easily be tested, such as editing colors, adding texts, and so on. All participants attempted to address issues raised by the tool. However, a few novice and intermediate participants found it difficult to address more complex problems, such as diverting or attracting attention in the saliency map. For example, P3 spent all their iterations to balance the salience of a chart, and P7 wanted the

salience to focus on a particular area.

Role for Track: Logbook. The tracking capabilities play as a role similar to a design “logbook.” Tracking changes per design iteration allowed some participants to easily catch up to where the design was left off. A notable observation was that it also functioned as an evaluation tool that measures whether there were improvements between the current and the past version. P3, whose main goal throughout the study was to alleviate salience in textual areas, said, *“I consistently checked the rate of salience in the tracking interface. The drop in numbers reassured me that I was on the right path. The positive feedback felt like a reward for the adjustments I had made.”*

On the other hand, there were also a few participants who did not use tracking. They uploaded the designs one after another, successively, without much delay. P2 was an example, noting *“Because I remember what needs to be changed, I did not look into the tool.”*

Criteria for Accepting Feedback. As one might expect, there were no participants that accepted all the feedback given by Visualizationary for their designs. Designers clearly cannot make changes based on what they cannot understand (or disagree with). For example, P7 did not understand how to create a bar chart using the dataset that they had. Second, the designer must also agree to some extent that the feedback would enhance the design. For example, some designers disregarded feedback on color vision deficiency when the visualization was merely intended for their own use. When the designer was unsure of whether there would be improvements, if the feedback was easily actionable, then they just tried it and took the feedback when it did.

General Assessment. Participants considered the tool as an assistant that provided expert-driven feedback. Many people originally thought that it would be difficult to provide insights, but were

surprised at its capability to help in visualization design. Furthermore, the use of ChatGPT created a sense of a “human” responding to the design, which made some participants less isolated during the design process.

However, some areas could be enhanced. One main comment was that despite the hierarchical presentation, several participants felt somewhat overwhelmed by the volume of text in the report. Some participants felt that with the help of visual images and text combined, this tool would generate more effective feedback.

5.5.3 Design Evaluation

We here present the results of the design assessment by external reviewers, which is the final step of our study procedure (see § 5.4.3). We recruited three senior visualization researchers who all have more than 7 years of experience to evaluate the designs conducted by the participants. Table 5.3 shows the overall ratings on the improvement of visualization designs conducted by our participants. Note that these values are not on the quality of the design but on the improvement of the design. On the whole, evaluators largely believed that the participants’ designs improved from their starting point, averaging a score of 3.69 (3 is neutral). P2, P7, and P12’s designs were modestly below the neutral mark. Evaluators observed consistency in P2’s designs, with a slight decrease in color quality. For P7, while adding a title was beneficial, the chart’s readability declined. P12 enlarged the chart for improved readability, but received mixed reviews about the choice of color. We also compare the improvements per different expert levels. We find that the mean score of improvements from the novice group is 3.34, from the intermediate group is 3.92, and from the expert group is 4.11. While we observe positive signs from all groups, we see more

improvements from intermediate and expert groups.

5.6 Discussion

The results from this study showed three things: that (1) a “vanilla” LLM, when provided with appropriate symbolic metrics and a “cheat sheet” of visualization design knowledge, can provide meaningful and actionable feedback on flaws in a visualization and how to improve them; that (2) even experienced visualization designers can benefit from this feedback; and that (3) the quality of visualization feedback is strongly dependent on the metrics extracted from the visualization. Below we discuss the potential of utilizing the system for novice designers. Then we enumerate limitations and avenues for future work.

5.6.1 On Visualization Design Feedback

Our main goal for Visualizationary was to help novice designers create visualizations. Our results indeed show that our tool is capable of detecting problems and providing relevant feedback according to existing design knowledge. However, it is also clear that participants with intermediate and expert knowledge used Visualizationary more effectively than novices. Based on comments from novice participants during the post-study interview, we speculate that automation and lack of foundational design skills could be contributing factors.

The tendency of novice users to rely on automated visualization suggestions, as offered by chart-authoring tools such as Microsoft Excel, contributes to this problem. This reliance caused some novices to lose their manual chart design skills (or never to acquire them in the first place). P12, who obtained a professional certificate for Microsoft Excel nearly a decade ago, said, “*the*

auto-generated charts are usually of good quality, so I don't feel the need to alter them." P11 shared similar comments. While automated tools have simplified visualization creation, their influence may have affected our results in Table 5.2.

Novices also have a harder time acting upon suggestions because they lack fundamental knowledge for visualization design. Many novice participants mentioned that when provided explanations and their corresponding suggestions, they still perceived it as high-level feedback; they preferred more actionable feedback where the impact of the suggested changes were clearly visible. In contrast, most participants with intermediate and expert levels understood the feedback well and used it to improve their work. P3, from the intermediate group, said, *"it was a good chance to identify that what I learned in textbooks really helps."* In fact, two experts (P1 and P9) noted that the lack of specific and actionable feedback could pose challenges for novices.

On the other hand, the ineffable purpose of design is to bring novel creation forth. While Visualizationary may currently fall short in providing detailed solutions, there is also danger in hampering the designer's creativity if suggestions become too prescriptive. The balance may need to shift depending on the experience of the designer, especially as the designer grows in experience. We leave this issue as future work.

5.6.2 Limitations and Future Work

While our work on Visualizationary in this paper builds on an entire stack of prior work, including the Scanner Deeply [6] virtual eye tracker and the Perceptual Pat [7] design tool, there are still weaknesses and potential shortcomings in our tool. We discuss them here.

Hallucination. Despite their capabilities, LLMs may produce nonsensical or unfaithful content, known as *hallucinations* [196]. Such hallucinations can lead to feedback that contradicts the designer’s original goals or does not improve the visualizations. This behavior potentially harms Visualizationary because users build trust in the models powering our tool when most of their outputs are truthful. Recent work proposes defensive methods to address hallucinations, e.g., incorporating a user’s feedback into the model’s fine-tuning [197, 198] and characterizing self-contradiction in model-generated texts [199, 200]. We leave the incorporation of these methods for future work.

Privacy risks. If we use commercial services built on LLMs, there is a *data privacy* concern about the user data [201]. A company may collect data—such as visualizations (images) and queries (text)—that a user provides to our tool and uses them to fine-tune their models further to improve the quality of their services. Later on, a motivated adversary can perform a series of privacy attacks on the fine-tuned models, such as membership/property inference attacks [202, 203] or sensitive data-extraction attacks [204, 205]. However, this only happens if our tool is based on LLMs hosted by a third-party. We can mitigate this concern by keeping all the LLMs we use in-house, e.g., by utilizing open-source models, such as Vicuna [179] and LLaMA [180], available on the Internet.

Towards effective design feedback. Visualizationary aims to improve visualization design by providing automated perceptual feedback. Our observations show that designers can easily apply feedback when the suggestions are straightforward. Regardless of their expertise levels, designers were able to address issues such as changing colors for color vision deficiency, clarifying

information, adding text, titles, or legends, and using highlighting techniques.

However, we also noted limitations in improving the design. The first issue arises when designers do not have a clear solution or are unsure about making changes. For example, several designers struggled with feedback from salience maps, which predict where viewers are likely to focus on charts. Designers often have different purposes and goals, making it difficult to adjust their designs to align with the salience map's indications. While there is research on understanding eye movement in visualization, there is little work assessing visualizations and providing guidance based on salience patterns. A more concrete assessment of such guidance could help designers evaluate their work and make informed changes.

The second limitation is that the impact of the feedback is constrained because visualization design may vary by target domain, and Visualizationary focuses on providing general feedback. For instance, creating an eye-catching visualization for commercial purposes might benefit from visual embellishments. To generate ideal feedback akin to that of an expert, users' intentions should be considered. This aspect, however, was missing in Visualizationary. We believe incorporating users' intents is the direction we should aim for to provide more advanced design recommendations in the future.

Future Work. Visualizationary paves the way for several new directions for future work on automating feedback for visualization design. First, a direct approach is to add filters to analyze visualization design for improved representation of a virtual human visual system. "Will visual embellishments in visualizations be identified as an object, or as a data representation by the viewer?" Answering such questions would help designers by providing more detailed, thorough feedback about their visualization designs. Another avenue for research might involve investigat-

ing the best methods—such as reports, conversational UIs that clarify vague suggestions textually, or tools offering visual examples—to deliver visualization feedback to various user groups.

An even more ambitious approach would be to expand the role of augmenting visualization design through feedback not just confined to iterative design process, but to every aspect of visualization design, including early design. We believe that the rapid development of text-to-image large language models could offer feedback that enhances designers’ creativity and expressiveness based on their intentions.

5.7 Conclusion

We have presented Visualizationary, a tool that leverages LLMs and visualization guidelines to provide design feedback for visualization designers. Visualizationary supports an automated visualization design workflow called ACGT (analyze-clarify-guide-track). We created a web-based interface to evaluate the effectiveness of Visualizationary and engaged 13 visualization designers of different seniority in a longitudinal design task. During this study, we conducted pre- and post-study interviews to further understand the designer’s experience using the tool. Finally, we asked three expert evaluators to assess their resulting designs. Overall, our results show that an off-the-shelf LLM can indeed provide high-quality and actionable feedback for novice, intermediate, and expert users alike on how to design a visualization.

Chapter 6: Conclusion

I conclude the dissertation by providing the summary, and future works about the thesis.

6.1 Summary

This dissertation focuses on creating a virtual human vision-inspired system to augment visualization designers in their design process. The goal was to create a system that provides guidance for visualization designers.

As the first step, I explored a methodology to instill human perception into machines by creating a virtual eyetracker for visualizations. Based on a taxonomy of 30 prior works on eye-tracking research, we found that collecting gaze with low-tech requirements using crowdsourcing and training the model with a deep learning approach are the *desiderata* for building a virtual eyetracker for visualizations. Following our desired approach, we built a virtual eyetracker called **A SCANNER DEEPLY**. We found that low-tech eyetrackers can often have noise, so we developed a technique to effectively remove noise from the eye movement data. By observing the patterns in Scanner Deeply, we also identified that the salience patterns are task- and chart-specific.

As the second step, I aimed to develop a virtual, human vision-inspired system that could provide design feedback for visualizations in an actionable and pragmatic manner. To that end, we developed a system called Perceptual Pat. Perceptual Pat uses a suite of computer vision filters

to provide design feedback, representing a human vision-inspired visual system in a pragmatic and actionable manner. For experimentation, we implemented **Perceptual Pat** on a web-based Pat Design Lab and allowed four experienced visualization designers to use the system. We found that Perceptual Pat helps visualization design by detecting issues in the designs.

Finally, as the last step, I aimed to automate the entire feedback generation process, from the analysis of visualizations to the delivery of interpretations and suggestions to the designers. This required interpreting insights from the analysis of chart images, generating corresponding suggestions, and using an effective text-processing technique to deliver the feedback to designers. To accomplish this, I developed a pipeline abbreviated as ACGT (analyze-clarify-guide-track). As an implementation, I developed a system called **Visualizationary**, which, upon receiving an image, provides design feedback about the image. Visualizationary uses ChatGPT, a vision-language model called DePlot, and a preamble of design knowledge about visualizations to provide suggestions and their corresponding interpretations to the designer. We found that feedback from Visualizationary improves the quality of visualizations for all levels of expertise, including experienced visualization designers.

6.2 Future Work

We can envision future work in two directions: first, further understanding the mechanisms behind human perception, and second, focusing on the human-centered use of AI for visualization design.

First, we can strive to discover more about understanding and instilling properties related to human perception. While our approach was to create a virtual eye tracker, numerous unanswered

questions remain about the mechanisms behind our eye movements when viewing visualizations. Many research areas are still uncovered. For example, eye movement patterns using a heatmap have not considered temporal changes. Additionally, we need to understand the relationship between tasks and eye movement patterns. Beyond imitating human eye movement, numerous types of biases persist in our perceptions, many of which are yet to be discovered. The most effective method to train these biases into a model to imitate an "ideal" human eye remains unknown. These topics represent future work in advancing how humans understand visualizations.

Second, we can focus on a human-centered use of AI for augmenting visualization design. Visualizationary succeeded in providing visualization feedback in the form of text, showing improvements regardless of the levels of expertise in visualizations. However, the results also indicate that it remains challenging for novices to make changes and improve the visualization during their iterative design process. Various research has identified the characteristics of visualization novices, and with existing techniques and chart authoring tools, novice designers lack the ability to control the chart and the knowledge to create effective charts. These limitations restrict the full potential of data visualization as an effective medium for design. This vision represents another possible area for future work.

Appendix A: Appendix – Visualizationary

In this section, I provide the prompts used for Visualizationary, and additional comparison of expert reviews of participants’ works with language models, such as ChatGPT, and WolframGPT.

A.1 Prompts used for Visualizationary

This section describes prompts used for Visualizationary. First, I describe the preamble, and then list additional prompts added for Visualizationary. The report Visualizationary consists of 53 questions, and each question consists of one prompt. For each question, we add to the prompt in the following manner: `[preamble] + [prompt-question] + [prompt-answer]`. We used ChatGPT-3.5-Turbo for the experiment.

A.1.1 Preamble used for Visualizationary

This part of the code added at the first part of the prompt:

Assume that you are a seasoned visualization expert, with more than 10 years of experience in data visualization. You are knowledgeable in data visualization, You are a person who looks at a data visualization and provides design feedback in the form of textual information. Out of many parts for visualization, you provide feedback from 5 perspectives: visual representation, textual content, color, visual saliency, and color vision deficiency. In visual

saliency, you analyze visual saliency and analyze patterns. Here, the primary goal is to simulate how people would read visualizations and warn designers of the possible biases that might occur for the readers. We provide three parts of the salience: center, text, and low-level stimuli. Most visual saliency models have center bias. They differ by the visualization image. First, we detect visualization images that overly focus at the center. This means that possibly, when people look into the chart image, it can warn people that the salience created by visualization may be focusing too much on the center/textual areas, or maybe focusing too little on low-level features. It is up to the designer to make the decision based on the information provided by the designer. In colors, we look at two things: whether there are similar colors and the variety of colors used for the visualization. The purpose of this is also to warn designers. First, when similar colors are used, we warn of the possibility of being ambiguous. Second, when a lot of different colors are used, we warn that if similar colors are used, then they can be interpreted as continuous values. In textual content, we first detect whether there is textual information in the chart. The goal is to inform users that textual information can help the understanding of the visualization. Second, we first detect whether there is a title in the visualization image. Title is an important component, so this is to encourage the addition of title if the title does not exist in the chart. Here, ChatGPT is used, along with information by APT by Jock MacKinlay, In color vision efficiency, we provide information on how the visualization would be seen by people with color vision deficiency. The goal is to simulate how people with color vision deficiency would view the image. Examples of color vision deficiency include protanopia, deuteranopia, and tritanopia. Designers can visually check how people with color vision deficiency would view the chart in the report. As a metric, we measure the loss of entropy occurring from the original to the translated image.

A.1.2 Prompts used for Visualizationary using ChatGPT

This section shows the prompts used for developing answers in Visualizationary. First, we provide questions asked for each visualization image. Each text within a citation mark refers to one question.

A.1.2.1 Questions, [prompt-question]

Part 0 Prompt questions for tracker mode: ‘Given the information about the changes, in one sentence, concisely and shortly, describe the changes made between the current and previous version about visual salience? Answer concisely, clearly and shortly, in one sentence. ’, ‘Given the information about the changes, in one sentence, concisely and shortly, describe the changes made between the current and previous version about text? Answer concisely, clearly and shortly, in one sentence. ’, ‘Given the information about the changes, in one sentence, concisely and shortly, describe the changes made between the current and previous version about visual representation? Answer concisely, clearly and shortly, in one sentence. Note that in all versions, chartjunk detector is run. If chartjunk detector is not mentioned in any of the versions, it just means that no chartjunk object was found. ’, ‘Are there changes in color composition? Given the information about the changes, in one sentence, concisely and shortly, describe the changes made between the current and previous version about color? Answer concisely, clearly and shortly, in one sentence. ’, ‘Given the information, in one sentence, concisely and shortly, describe the changes made between the current and previous version about accessibility? Answer concisely, clearly and shortly, in one sentence.’”

Part 1 ‘Analyze the visual salience on text. Provide interpretations shortly, in one sentence.’, ‘Provide suggestions based on the previous question. Answer shortly, in one sentence.’, ‘Sum up the responses in previous, and the previous questions about salience on text. Provide the answer clearly and concisely.’, ‘Analyze the visual salience on the center of the chart image. Provide interpretations shortly, in one sentence.’ , ‘Provide suggestions based on the results of the previous question. Answer shortly, in one sentence.’, ‘Sum up the responses in previous, and the previous questions about salience on the center zone in the image. Provide the answer clearly and concisely.’,

‘Analyze the visual salience on low-level visual stimuli. Provide interpretations. Answer shortly, in one short sentence.’, ‘Provide suggestions based on the results of the previous question. Answer shortly, in one sentence.’, ‘Sum up the responses in the previous, and the previous question about salience on low-level visual stimuli. Provide the answer clearly and concisely.’, ‘Sum up the interpretations made about visual salience on text, center, and low-level visual stimuli only by judging from responses in previous questions. Provide the answer, concisely, in one sentence.’, ‘Sum up the suggestions made about visual salience on text, center, and low-level visual stimuli only by judging from responses about previous questions. Provide the answer, concisely, in one sentence.’, ‘Provide a short, general overview about the results from all questions about visual salience. Discuss only the most important points. Give the response concisely, in one sentence.’,

Part 2 ‘Is the title detectable from the chart image? Provide the answer shortly, in one sentence. Look at the Deplot Table to look for the title.’, ‘If the title is detected, can you provide suggestions about improving the title? Answer concisely and shortly, in one sentence.’, ‘If the title is detected, can you evaluate the title?’, ‘Sum up the responses in previous, and the previous questions about title. Provide the answer clearly and concisely.’, ‘Is any textual content detectable from the chart image? Provide the answer shortly, in one sentence. In answering the question, refer to the OCR text.’, ‘Provide suggestions based on the results of the previous question. Answer shortly, in one sentence.’, ‘Sum up the responses in previous, and the previous questions about textual content in one sentence. Provide the answer clearly and concisely.’, ‘Sum up the interpretations made about title and textual content only by judging from your responses in previous questions. Provide the answer, concisely, in one sentence.’, ‘Sum up the suggestions made about title and textual content only by judging from your responses in previous questions. Provide the answer, concisely, in one sentence.’, ‘Provide a short, general overview about the results from all questions about text. Discuss only the most important points. Give the response concisely, in one sentence.’,

Part 3 ‘According to Deplot, is the data table detectable in the chart image? Provide your answer shortly, in one sentence.’, ‘Based on the data table, what seems the optimal chart to describe the patterns, distributions in the data table? Provide your answer concisely and shortly, in one sentence.’, ‘Can you explain the reasons for choosing the following chart in the previous question? Provide your answer concisely and shortly.’, ‘Sum up the responses in previous, and the previous questions about the data table and the optimal chart. Provide the answer

clearly and concisely.’, ‘According to the Chartjunk filter, are any chartjunk/visual embellishments identifiable in the chart image? Answer shortly and concisely, in one sentence.’, ‘Provide suggestions based on the results of the previous question, about chartjunk/visual embellishments. Answer shortly and concisely, in one sentence.’, ‘Sum up the responses in previous, and the previous questions about chartjunk/visual embellishments, Provide the answer clearly, shortly, and concisely.’, ‘Sum up the interpretations made about the data table and the optimal chart only by judging from your responses in previous questions. Provide the answer shortly and concisely, in one sentence.’, ‘Sum up the suggestions made about the data table and the optimal chart only by judging from your responses in previous questions. Provide the answer shortly and concisely, in one sentence.’, ‘Provide a short, general overview about the results from all questions about visual representations. Discuss only the most important points. Give the response concisely, in one sentence.’,

Part 4 ‘According to the color filter, how many colors are used in the chart image? Answer shortly, in one sentence.’, ‘Provide suggestions about the previous question. Answer concisely and shortly, in one sentence.’, ‘Provide detailed utility of colors in data visualization under different types of data.’, ‘Sum up the responses in previous, and the previous questions about the variability of colors. Provide the answer clearly and concisely.’, ‘According to the color filter, are some colors similar? Provide the answer in one sentence.’, ‘Provide suggestions about the result of the previous question. Answer shortly, in one sentence.’, ‘Sum up the responses in previous, and the previous questions about similar colors. Provide the answer concisely and shortly, in one sentence.’, ‘Sum up the interpretations made about the variability of colors and similar colors only by judging from your responses in previous questions. Provide the answer shortly and concisely, in one sentence.’, ‘Sum up the suggestions made about the variability of colors and similar colors only by judging from your responses in previous questions. Provide the answer shortly and concisely, in one sentence.’, ‘Provide a short, general overview about the results from questions about colors? Discuss only the most important points. Give the response concisely, in one sentence.’,

Part 5 ‘Is there information loss when people with Deuteranopia see the chart image? Provide interpretations. Answer shortly, in one sentence.’, ‘Provide suggestions about the result of the previous question. Answer concisely and shortly, in one sentence.’, ‘Sum up the responses in previous, and the previous questions about Deuteranopia.

Provide the answer clearly and concisely.’, ‘Is there information loss when people with Protanopia see the chart image? Provide interpretations. Answer shortly, in one sentence.’, ‘Provide suggestions about the result of the previous question. Answer concisely and shortly, in one sentence.’, ‘Sum up the responses in previous, and the previous questions about Protanopia, concisely. Answer clearly, shortly, and concisely.’, ‘Is there information loss when people with Tritanopia see the chart image? Provide interpretations. Answer shortly, in one sentence.’, ‘Provide suggestions about the result of the previous question. Answer concisely and shortly, in one sentence.’, ‘Sum up the responses in previous, and the previous questions about Tritanopia, concisely. Answer clearly, shortly, and concisely.’, ‘Sum up the interpretations made about color vision deficiency (Deuteranopia, Protanopia, Tritanopia) only by judging from your responses in previous questions. Provide the answer shortly and concisely, in one sentence.’, ‘Sum up the suggestions made about color vision deficiency (Deuteranopia, Protanopia, Tritanopia) only by judging from your responses in previous questions. Provide the answer shortly and concisely, in one sentence.’, ‘Provide a short, general overview about the results from questions in Part 5? Discuss only the most important points. Give the response concisely, in one sentence.’,

A.1.2.2 Answers, [prompt-answer]

This contains the suggestions referred as [filter-suggestions]

Part 1 ‘The question is ”Analyze the visual salience on text. Provide interpretations shortly, in one sentence.” Please interpret exactly in the following way: If no visual salience is detected, then just interpret as ”No salience is detected in the chart image.” If (the rate of salience in textual zone over the rate of textual zone in chart image) from the measured result is less than 0.6 , then interpret it as ”there is a lack of salience in textual elements.” If the measured result is between 0.6 and 2.0, just interpret that ”there is moderate amount of salience in text.”, else, if (the rate of salience in textual zone over the rate of textual zone in chart image) from the measured result is more than 2.0, then interpret it as ”there is an overabundance of salience in textual elements.”’, ‘The question is ”Provide

suggestions based on the previous question. Answer shortly, in one sentence.” Please provide suggestions exactly in the following way, and try to keep the answer concise: If the answer to the previous question is ”No salience is detected in the chart image,” then do not give any suggestions, or say ”there is not suggestion to give”. Else, if the answer to the previous question is ”there is a lack of salience in textual elements,” then suggest by saying ”highlight where it seems to be important, such as using colors with high visual stimuli, or changing the font of textual components, or changing the color, shape of visual components.” Else, if the answer to the previous question is ”there is an overabundance of salience in textual zone of the chart image,” then warn the possibility that the text can override the visualization, and suggest to alleviate the emphasis on textual information. Else, if there is ”a moderate amount of salience in textual zone of the chart image”, then do not provide any specific suggestions.’, ‘nothing’, ‘The question is ”Analyze the visual salience on the center of the chart image. Provide interpretations shortly, in one sentence.” Please interpret exactly in the following way: If (the rate of salience in central zone over the rate of central zone in chart image) is less than 2.2, then interpret as ”there seems to be enough focus of salience in the center zone.” Else, if (the rate of salience in central zone over the rate of central zone in chart image) is more than 2.2, then interpret it as ”there is an abundant concentration of salience in center zone.”’, ‘The question is ”Provide suggestions based on the results of the previous question. Answer shortly, in one sentence.”. Please provide suggestions exactly in the following way, and try to keep the answer concise. If the answer to the previous question is ”there is an abundant concentration of salience in center zone.”, then warn the possibility that the salience of the visualization is overly focused towards the center, because there is not much areas within the chart image that can be the point of interest, such as lack of title, legend, or textual information. People may not know what to look for when viewing charts. Hence, suggest to add or highlight such components. If the answer to the previous question is ”there seems to be enough focus of salience in the center zone” then do not provide any particular suggestions.’, ‘nothing’, ‘The question is ”Analyze the visual salience on low-level visual stimuli. Provide interpretations. Answer shortly, in one short sentence.” Please interpret exactly in the following way: If the (rate of salience in zone with low-level visual stimuli over the rate of zone with low-level visual stimuli in chart image) is less than 0.8, then interpret it as ”the chart image has a lack of salience in zones with low-level visual stimuli.” Else, then interpret it as ”the chart image does not a lack of salience in zones with low-level visual stimuli.”’, ‘The question is ”Provide suggestions based on the results of the previous question. Answer shortly, in one sentence.” Please try to keep the answer concise. If the

answer is "the chart image has a lack of salience in zones with low-level visual stimuli.", then suggest methods to highlight parts that deserves to attract attention, such as highlight a part of visual cues with colors, or make the text bigger. Else if the image has an abundance of visual salience, it is fine, so do not give any specific suggestions.', 'nothing', 'nothing', 'nothing', 'nothing',

Part 2 'Given an image, the question is "Is the title detectable from the chart image? Provide the answer shortly, in one sentence. Look at the Deplot Table to look for the title.". Please answer in the following manner. You can find to this question from the table, so look closely at the table. Interpret in the following manner: If there is a title in chart image, then interpret this as "there is title in the visualization." If, however, there is not a title in the chart image, then interpret as "the title is not detected."', 'The question is "If the title is detected, can you provide suggestions about improving the title? Answer concisely and shortly, in one sentence." Please answer in the following manner. If the answer to the previous question is that "there is title in visualization," then suggest that a good visualization should at least have labels, titles, and tick marks. If the answer to the previous question is that "the title is not detected," then suggest that the choice of semantic content and placement of annotations can influence what the readers take away from the chart and how likely they are to use the annotations in their takeaways.', 'The question is "If the title is detected, can you evaluate the title?" Please answer in the following manner: If the title is not detected, then do not make any comments to this question. However, if the title is detected, mention the following: "These are the factors that lead to a good title. These are (1) features importance (title should convey key features), (2) conciseness (a concise title allows users to understand quickly), (3) preciseness (preciseness measures how closely and accurately title represents data), (4) generic information richness (data attributes and data range), (5) coverage (it is important not to omit details for comprehensive information), and (6) non-technicality (non-technicality refers to the degree of specialized knowledge or background required to understand a visualization's title. The number of words and the complexity of words is the most important factor for the non-technicality of visualization titles.) If possible, make comments about the title from these factors." Also, if the title is detected, make judgement about the title based on the mentioned factors that lead to a good title (these are: features importance, conciseness, preciseness, generic information richness, and coverage).', 'nothing', 'The question is "Is any textual content detectable from the chart image? Provide your answer shortly, in one sentence. In answering the question, refer to the OCR text." You can

find whether there is title or not from results from the OCR filter of the chart image. Provide the answer in the following manner: If there is any textual content in chart image, then answer in the following way: "there is textual content in the chart image." If there is not a textual content in the chart image, then answer in the following way: "there is no textual content in the chart image."', 'The question is "Provide suggestions based on the results of the previous question. Answer shortly, in one sentence." Please answer in the following manner: If the answer to the previous question is "there is textual content in the chart image," then suggest methods to better position these textual elements. These are: G1: Rather than aiming for maximally minimalist design, annotate charts with relevant text. G2: The best semantic content depends on the intended message. G3: To convey a message, the best placement of the text depends on the semantic content of the text. G4: Consider a text-only variant that can stand alone. If the question is interpreted as having no textual content, then suggest that adding texts to visualization can help better understand charts.', 'nothing', 'nothing', 'nothing', 'nothing',

Part 3 'The question is "According to Deplot, is the data table detectable in the chart image? Provide your answer shortly, in one sentence." Please answer in the following manner: You can find whether there the data table is detectable or not from the data table created by the Deplot. Provide the answer in the following manner: If there is a data table from Deplot, then answer this as "the data table is detectable". However, if there is not a data table from Deplot, then interpret as "the data table is not detectable."', 'The question is: "Based on the data table, what seems the optimal chart to describe the patterns, distributions in the data table? Provide your answer concisely and shortly, in one sentence." Please answer in the following manner: If the data table is not detected, then do not make any suggestions. However, if the data table is detected, then try to find the optimal chart that can best describe the patterns and distributions in the data table, based on these encoding techniques below: "The primitive encoding technique for single-position are (prioritized, from the left) horizontal axis and vertical axis. The primitive encoding technique for apposed position are (prioritized, from the left) line chart, bar chart, and plot chart. The primitive encoding technique for map (prioritized, from the left) are road map, topographic map. The primitive encoding technique for connection (prioritized, from the left) are tree, acyclic graph, network. The primitive encoding technique for misc. (prioritized, from the left) are pie chart, venn diagram, etc.." Then, suggest the optimal chart, and briefly provide the reasons.', 'The question is: "Can you explain the reasons for choosing the

following chart in the previous question? Provide your answer concisely and shortly.” and provide reasons for the answer to the previous question. The evaluation should be done based on this: ”The primitive encoding technique for single-position are (prioritized, from the left) horizontal axis and vertical axis. The primitive encoding technique for apposed position are (prioritized, from the left) line chart, bar chart, and plot chart. The primitive encoding technique for map (prioritized, from the left) are road map, topographic map. The primitive encoding technique for connection (prioritized, from the left) are tree, acyclic graph, network. The primitive encoding technique for misc. (prioritized, from the left) are pie chart, venn diagram, etc..”, ‘nothing’, ‘The question is: ”According to the Chartjunk filter, are any chartjunk/visual embellishments identifiable in the chart image? Answer shortly and concisely, in one sentence.” This is detected by the Chartjunk filter. Please answer in the following manner: If the chartjunk filter answers as ”One or more chartjunk/Visual embellishment object is detected in the chart image.”, then answer as ”Chartjunks/visual embellishments are detected.” On the other hand, if the chartjunk filter answers as ”No chartjunk/Visual embellishment is detected in the chart image.”, then answer as ”Chartjunk/visual embellishments were not detected.”’, ‘The question is: ”Provide suggestions based on the results of the previous question, about chartjunk/visual embellishments.” Please answer in the following manner: If the answer from the previous question is ”Chartjunks/visual embellishments are detected”, then suggest the following: ”The addition of pictorial elements that embellish data with anthropomorphic or metaphorical elements – intended to enhance engagement or memory – has been demonized as chartjunk.” At the same time, mention the following fact: ”Despite mixed evidence as to whether their presence improves memory for the data, pictorial elements do improve memory for the fact that a visualization was previously seen, both in the short and the longer term.” However, if the answer from the previous question is ”Chartjunk/visual embellishments were not detected,” then suggest the following: ”Despite mixed evidence as to whether their presence improves memory for the data, pictorial elements do improve memory for the fact that a visualization was previously seen, both in the short and the longer term” and ”Given the limited working memory resources of visualization viewers, designers often recommend a minimalist aesthetic that strips away any design elements that are not critically needed” and ”Researchers who have found the opposite have typically argued either that viewers’ higher level of familiarity with cluttered charts make those charts more attractive or that decluttered charts that are too minimalistic become boring.”’, ‘nothing’, ‘nothing’, ‘nothing’, ‘nothing’,

Part 4 ‘The question is “According to the color filter, how many colors are used in the chart image? Answer shortly, in one sentence.” The answer will be provided by the color filter. Provide the answer in the following way: If the chart has no color, then simply answer as “the chart has no colors.” If the filter detects a list of colors, then answer that “the chart contains multiple colors” Also, list the colors.’, ‘The question is, “Provide suggestions about the previous question. Answer concisely and shortly, in one sentence.” If the answer to the previous question is “the chart contains multiple colors”, then suggest the following to warn the designers of using a lot of colors: “Color can be used for ordinal measurements, but that full color spectrum is not ordered. However, parts of the color spectrum are ordinally perceived.” However, if the answer to the previous question is “the chart has no colors,” then make the following suggestions: “Colors can make the data representation of various data (quantitative, ordinal, or nominal) perceptually salient. Their effectiveness vary depending on the data represented.”’, ‘The question is: “Provide detailed utility of colors in data visualization under different types of data.” Answer in the following manner: “Colors can effectively represent information, but their effectiveness differ. Here are the rankings of quantitative perceptual tasks for visualizations (the list is ordered, from the strongest to weakest): Position, Length, Angle, Slope, Area, Volume, Density, Color Saturation, Color Hue, Texture, Connection, Containment and Shape are not relevant to these types of data. Here are the rankings of ordinal perceptual tasks for visualizations (the list is ordered, from the strongest to weakest): Position, Density, Color Saturation, Color Hue, Texture, Connection, Containment, Length, Angle, Slope, Area, Volume. Shape is not relevant to this type of data. Here are the rankings of nominal perceptual tasks for visualizations (the list is ordered, from the strongest to weakest): Position, Color Hue, Texture, Connection, Containment, Density, Color Saturation, Shape, Length, Angle, Slope, Area, Volume.”’, ‘nothing’, ‘The question is, “According to the color filter, are some colors similar? Provide the answer in one sentence.” Whether the chart has similar colors will be detected by the color filter. Please answer in the following manner: If there are similar colors in the chart image, then answer as “there are similar colors in the chart image” and list the similar colors. However, if no similar colors are detected in the chart image, then interpret as “No similar colors are detected.”’, ‘The question is, “Provide suggestions about the result of the previous question. Answer shortly, in one sentence.” If the answer to the previous question is “there are similar colors in the chart image”, then suggest the following, “When continuous values are encoded through visual channels, those values can be warped by categorical perception. One example is the rainbow. The same phenomenon occurs when data are depicted by hues.” However, if the

answer to the previous question is "No similar colors are detected.", then do not provide any suggestions.', 'nothing', 'nothing', 'nothing', 'nothing',

Part 5 "The question is, "Is there information loss when people with Deuteranopia see the chart image? Provide interpretations. Answer shortly, in one sentence." Provide the answer in the following manner: If the information loss of the chart image after changing to how people with Deuteranopia would see is more than or equal to 0.392, then answer as "There is noticeable information loss in the chart image when converted to how people with Deuteranopia would see". However, if the information loss of the chart image after changing to how people with Deuteranopia would see is less than 0.392, then answer as "There is minor information loss in the chart image when converted to how people with Deuteranopia would see".', 'The question is, "Provide suggestions about the result of the previous question. Answer concisely and shortly, in one sentence." Please answer in the following manner: If answer to the previous question is "There is minor information loss in the chart image when converted to how people with Deuteranopia would see," then do not make any specific suggestions. However, if the answer to the previous question is "There is noticeable information loss in the chart image when converted to how people with Deuteranopia would see," then provide the following suggestion: "Use a colorblind-friendly palette when appropriate. For example, blue/orange is a common colorblind-friendly palette. Blue/red or blue/brown would also work. For the most common conditions of color-vision deficiencies, all of these work well, since blue would generally look blue to someone with color vision deficiencies. (taken from www.tableau.com)."', 'nothing', 'The question is, "Is there information loss when people with Protanopia see the chart image? Provide interpretations. Answer shortly, in one sentence." Please interpret in the following manner: If the information loss of the chart image after changing to how people with Protanopia would see is more than or equal to 0.374, then interpret as "There is noticeable information loss in the chart image when converted to how people with Protanopia would see". However, if the information loss of the chart image after changing to how people with Protanopia would see is less than 0.374, then interpret as "There is

minor information loss in the chart image when converted to how people with Protanopia would see”.’, ‘The question is, “Provide suggestions about the result of the previous question. Answer concisely and shortly, in one sentence.” Please answer in the following manner: If answer to the previous question is “There is minor information loss in the chart image when converted to how people with Protanopia would see,” then do not make any specific suggestions. However, if the question is answered as “There is noticeable information loss in the chart image when converted to how people with Protanopia would see,” then provide the following suggestion: “Use a colorblind-friendly palette when appropriate. For example, blue/orange is a common colorblind-friendly palette. Blue/red or blue/brown would also work. For the most common conditions of color-vision deficiencies, all of these work well, since blue would generally look blue to someone with color vision deficiencies. (taken from www.tableau.com).”’, ‘nothing’, ‘The question is, “Is there information loss when people with Tritanopia see the chart image? Provide interpretations. Answer shortly, in one sentence.” Please interpret in the following manner: If the information loss of the chart image after changing to how people with Tritanopia would see is more than or equal to 0.355, then interpret as “There is noticeable information loss in the chart image when converted to how people with Tritanopia would see”. However, if the information loss of the chart image after changing to how people with Tritanopia would see is less than 0.355, then interpret as “There is minor information loss in the chart image when converted to how people with Tritanopia would see”.’, ‘The question is, “Provide suggestions about the result of the previous question. Answer concisely and shortly, in one sentence.” If the answer to the previous question is “There is minor information loss in the chart image when converted to how people with Tritanopia would see,” do not make any specific suggestions. However, if the question is answered as “There is noticeable information loss in the chart image when converted to how people with Tritanopia would see,” then provide the following suggestion: “Use a colorblind-friendly palette when appropriate. For example, blue/orange is a common colorblind-friendly palette. Blue/red or blue/brown would also work. For the most common conditions of color-vision deficiencies, all of these work well, since blue would generally look blue to someone with color vision deficiencies. (taken from www.tableau.com).”’, ‘nothing’, ‘nothing’, ‘nothing’, ‘nothing’

A.1.2.3 Notes

Note that the values from the filter are added as prompts, too in the system. References used to create suggestions: This code used metrics from Liu et al.'s Autotitle [206] in recommending the desired title. The preamble for chart recommendations are based on texts from MacKinlay's APT [195]. For various feedback on perception on visualizations, we referenced Franconeri's survey on visual datacommunication [121].

A.1.2.4 Examples of outputs

We present two responses generated by ChatGPT for Visualizationary report. While there exists evidence that the questions are generated using an existing template, from our experiment we did not observe hallucination from the experiment, or our case study. Furthermore, we empirically ran Visualizationary 10 times with the same image as the input, and were not able to find hallucination issues in them.

First example '0 There is ample amount of salience in textual elements.', '1 There is not suggestion to give.', '2 The presence of salience in textual elements is ample.', '3 There is an abundant concentration of salience in the center zone.', '4 There seems to be enough focus of salience in the center zone, no particular suggestions needed.', '5 Based on the responses, it can be concluded that the central zone in the image has enough concentration of salience.', '6 The chart image does not have a lack of salience in zones with low-level visual stimuli.', '7 If the chart image has a lack of salience, consider highlighting important parts with contrasting colors or larger text.', '8 The responses indicated that highlighting important parts with contrasting colors or larger text can improve salience on low-level visual stimuli.', '9 Visual salience is present in textual elements, concentrated in the center zone, and not lacking in low-level visual stimuli.', '10 Enhance salience on low-level visual stimuli by using contrasting colors or larger text to highlight important parts.', '11 Overall, the responses regarding visual salience suggest a general sufficiency of

salience in the center zone of images and text elements, with no specific recommendations needed for improvement.’, ‘12 Yes, the title is detectable from the chart image as ”Government spending on pension benefits % of GDP”.’, ‘13 If the title is detected, suggest that a good visualization should at least have labels, titles, and tick marks.’, ‘14 These are the factors that lead to a good title for the title ”Government spending on pension benefits % of GDP”: 1) Features importance: The title conveys the key feature of government spending on pension benefits as a percentage of GDP. 2) Conciseness: The title is concise and allows users to understand the information quickly. 3) Preciseness: The title is precise in representing the specific data point being highlighted. 4) Generic information richness: The title provides specific data attributes (% of GDP) and the topic (government spending on pension benefits). 5) Coverage: The title covers the important details related to government spending on pension benefits in relation to GDP. 6) Non-technicality: The title may require some background knowledge about GDP and government spending, but it is not overly technical.’, ‘15 Overall, the title ”Government spending on pension benefits % of GDP” is effective based on the factors that lead to a good title.’ ‘16 The title ”Government spending on pension benefits % of GDP” is effective as it conveys key features, is concise and precise, provides generic information richness, covers important details, and is non-technical.’, ‘17 There is textual content in the chart image.’, ‘18 Adding text to the visualization can help improve understanding of the charts.’, ‘19 The responses indicated that adding textual content to the chart image can enhance comprehension and clarity of the information presented.’, ‘20 The title is ”Government spending on pension benefits % of GDP” and there is textual content in the chart image.’, ‘21 Ensure the title is concise, precise, rich in information, and non-technical, and include textual content to further enhance understanding of the visualization.’, ‘22 The title ”Government spending on pension benefits % of GDP” effectively conveys key features, is concise and precise, provides generic information richness, covers important details, and is non-technical, establishing it as an effective title for the visualization.’, ‘23 the data table is detectable.’, ‘24 The optimal chart to describe the patterns and distributions in the data table is a bar chart because it allows for easy comparison of the data values across categories, making it ideal for displaying the information in a clear and concise manner.’, ‘25 The reason for choosing a bar chart in the previous question is because it is the optimal chart to describe the patterns and distributions in the data table. Bar charts allow for easy comparison of data values across categories, making it ideal for displaying information in a clear and concise manner.’, ‘26 The optimal chart to describe the patterns and distributions in the data table is a bar chart because it allows for easy comparison of the data values across categories,

making it ideal for displaying the information in a clear and concise manner.’ ‘27 Chartjunk/visual embellishments were not detected.’, ‘28 Despite mixed evidence as to whether their presence improves memory for the data, pictorial elements do improve memory for the fact that a visualization was previously seen, both in the short and the longer term” and ”Given the limited working memory resources of visualization viewers, designers often recommend a minimalist aesthetic that strips away any design elements that are not critically needed” and ”Researchers who have found the opposite have typically argued either that viewers’ higher level of familiarity with cluttered charts make those charts more attractive or that decluttered charts that are too minimalistic become boring.’, ‘29 Chartjunk/visual embellishments should be used sparingly in data visualizations, as they can help improve memory for the data but may also overwhelm viewers with unnecessary elements. It is recommended to maintain a minimalist aesthetic to ensure that the design elements do not detract from the main message of the visualization.’, ‘30 The data table is clear and informative, and the optimal chart is free of unnecessary visual embellishments.’, ‘31 A bar chart is recommended as the optimal chart to display the data table patterns and distributions clearly and concisely.’, ‘32 Overall, the data showed that bar charts are the optimal choice for displaying patterns and distributions in a clear and concise manner, while the presence of pictorial elements can improve memory for a visualization in both the short and longer term. It is important to consider working memory limitations and use a minimalist aesthetic to avoid overwhelming viewers.’, ‘33 The chart contains multiple colors.’, ‘34 The previous question suggests providing suggestions based on whether the chart contains colors or not, emphasizing the importance of using colors effectively for different types of data representation.’, ‘35 Colors can effectively represent information, but their effectiveness differ. Here are the rankings of quantitative perceptual tasks for visualizations (the list is ordered, from the strongest to weakest): Position, Length, Angle, Slope, Area, Volume, Density, Color Saturation, Color Hue, Texture, Connection, Containment and Shape are not relevant to these types of data.’, Here are the rankings of ordinal perceptual tasks for visualizations (the list is ordered, from the strongest to weakest): Position, Density, Color Saturation, Color Hue, Texture, Connection, Containment, Length, Angle, Slope, Area, Volume. Shape is not relevant to this type of data.’, ‘36 Here are the rankings of nominal perceptual tasks for visualizations (the list is ordered, from the strongest to weakest): Position, Color Hue, Texture, Connection, Containment, Density, Color Saturation, Shape, Length, Angle, Slope, Area, Volume. Colors can effectively represent information, but their effectiveness differs based on the type of data being visualized. Position is the strongest perceptual task for all types of data, while color saturation and

color hue are more effective for representing nominal data. Shape is not relevant for quantitative or ordinal data.’, ‘37 There are similar colors in the chart image.’, ‘38 There are similar colors in the chart image. Suggestion: When continuous values are encoded through visual channels, those values can be warped by categorical perception. One example is the rainbow. The same phenomenon occurs when data are depicted by hues.’, ‘39 There are similar colors in the chart image which may result in data being interpreted inaccurately, particularly when using colors that are close to each other on the color spectrum such as in a rainbow.’, ‘40 The chart contains multiple colors, including similar colors.’, ‘41 Ensure that when using colors to represent data, be mindful of how similar colors can be perceived differently, especially when depicting continuous values using hues.’, ‘42 Colors can effectively represent information in visualizations, but their effectiveness can differ based on the type of data being represented and the visual tasks involved.’, ‘43 There is minor information loss in the chart image when converted to how people with Deuteranopia would see.’, ‘44 There is noticeable information loss in the chart image when converted to how people with Deuteranopia would see. Use a colorblind-friendly palette when appropriate. For example, blue/orange is a common colorblind-friendly palette. Blue/red or blue/brown would also work. For the most common conditions of color-vision deficiencies, all of these work well, since blue would generally look blue to someone with color vision deficiencies. (taken from www.tableau.com).’, ‘45 The responses suggest that there is information loss in an image when converted to how people with Deuteranopia would see. It is recommended to use a colorblind-friendly palette, such as blue/orange, blue/red, or blue/brown, to ensure better visibility for individuals with color vision deficiencies like Deuteranopia.’, ‘46 There is minor information loss in the chart image when converted to how people with Protanopia would see.’, ‘47 Use a colorblind-friendly palette when appropriate. For example, blue/orange is a common colorblind-friendly palette. Blue/red or blue/brown would also work. For the most common conditions of color-vision deficiencies, all of these work well, since blue would generally look blue to someone with color vision deficiencies.’, ‘48 Responses from previous questions suggest that there is minor information loss in charts when viewed by people with Protanopia. The suggestion is to use colorblind-friendly palettes such as blue/orange, blue/red, or blue/brown to make the content more accessible. These palettes work well for common color-vision deficiencies and ensure that the information is clear and easily understood by all users.’, ‘49 There is noticeable information loss in the chart image when converted to how people with Tritanopia would see.’, ‘50 There is minor information loss in the chart image when converted to how people with Tritanopia would see.’, ‘51 The responses

from previous questions about Tritanopia suggest that there is noticeable information loss in the chart image when converted to how people with Tritanopia would see it, with a minor information loss mentioned as well.’ ‘52 Color vision deficiencies, including Deuteranopia, Protanopia, and Tritanopia, result in varying degrees of information loss when viewing images, with each condition affecting color perception differently.’ ‘53 Use colorblind-friendly palettes such as blue/orange, blue/red, or blue/brown to accommodate individuals with color vision deficiencies such as Deuteranopia, Protanopia, and Tritanopia.’

Second example ‘0 There is an overabundance of salience in textual elements.’ ‘1 Highlight where it seems to be important, such as using colors with high visual stimuli, or changing the font of textual components, or changing the color, shape of visual components.’ ‘2 The key to addressing salience in text is to strategically highlight important elements using colors, fonts, and visual cues.’ ‘3 There is an overabundant concentration of salience in the center zone.’ ‘4 There is the possibility that the salience of the visualization is overly focused towards the center. This is because there is not much areas within the chart image that can be the point of interest, such as lack of title, legend, or textual information. People may not know what to look for when viewing charts. Hence, try to add or highlight such components.’ ‘5 The central zone in the image has an overabundant concentration of salience. It is suggested to add or highlight components such as a title, legend, or textual information to provide clear points of interest for viewers.’ ‘6 The chart image does not lack salience in zones with low-level visual stimuli.’ ‘7 Highlight parts that deserve to attract attention, such as emphasizing key data points or using contrasting colors.’ ‘8 Salience is key on low-level visual stimuli to attract attention to important elements. Make sure to highlight key areas and use contrasting colors to increase visibility.’ ‘9 Visual salience is present in an overabundance in textual elements, in the center zone, and in zones with low-level visual stimuli.’ ‘10 Enhance visual salience in text, center, and low-level visual stimuli by using colors, fonts, and shapes effectively to draw attention to key components and information.’ ‘11 The results show that there is an overabundance of salience in textual elements and a lack of salience in zones with low-level visual stimuli in chart images. It is important to highlight important areas and balance visual stimuli to guide viewers’ attention effectively.’ ‘12 There is a title in the visualization: ”Characteristics of dogs registered with the UK’s Kennel Club”.’ ‘13 If the title is not detected, suggest that the choice of semantic content and placement of annotations can influence what the readers take away from the chart.’ ‘14 These are the factors that lead to a good title. These are (1) features

importance (title should convey key features), (2) conciseness (a concise title allows users to understand quickly), (3) preciseness (preciseness measures how closely and accurately title represents data), (4) generic information richness (data attributes and data range), (5) coverage (it is important not to omit details for comprehensive information), and (6) non-technicality (non-technicality refers to the degree of specialized knowledge or background required to understand a visualization's title. The number of words and the complexity of words is the most important factor for the non-technicality of visualization titles.)

Judgement about the title based on the mentioned factors: - Features Importance: The title conveys the key feature of the data being about characteristics of dogs registered with the UK's Kennel Club. - Conciseness: The title is concise and to the point. - Preciseness: The title accurately represents the data being presented. - Generic Information Richness: The title includes information about the source of the data (UK's Kennel Club). - Coverage: The title provides a good overview of the data being presented. - Non-Technicality: The title is non-technical and can be easily understood by a wide audience.'

'15 The title "Characteristics of dogs registered with the UK's Kennel Club" meets the criteria for a good title, as it is concise, precise, rich in generic information, covers important details, and is non-technical.'

'16 There is textual content in the chart image.'

'17 If the previous question results indicate that the text content is not clear enough in the chart image, consider adding annotations or labels to make the information more comprehensive.'

'18 The responses indicate that the textual content in the chart image may not be clear and adding annotations or labels could make the information more comprehensive.'

'19 The visualization titled "Characteristics of dogs registered with the UK's Kennel Club" includes textual content.'

'20 The suggested title for the chart should convey key features, be concise, precise, rich in generic information, provide good coverage, and be non-technical, while annotations or labels can be added to make the textual content more comprehensive.'

'21 The title of the visualization "Characteristics of dogs registered with the UK's Kennel Club" effectively conveys key features, is concise, precise, includes information about the data source, provides a good overview, and is non-technical for easy understanding by a wide audience.'

'22 The data table is detectable.'

'23 A bar chart would be the optimal chart to describe the patterns and distributions in the data table as it can easily compare the values across different categories and identify any trends or relationships.'

'24 The reason for choosing a bar chart in the previous question is because it is a primitive encoding technique for apposed position, which is prioritized for comparing values across different categories and identifying trends or relationships in the data table.'

'25 Choosing a bar chart

as the optimal chart for the provided data table is recommended because it allows for easy comparison of values across different categories and helps in identifying trends or relationships within the data.’ ‘26 Chartjunk/visual embellishments were not detected.’ ‘27 ”Despite mixed evidence as to whether their presence improves memory for the data, pictorial elements do improve memory for the fact that a visualization was previously seen, both in the short and the longer term” and ”Given the limited working memory resources of visualization viewers, designers often recommend a minimalist aesthetic that strips away any design elements that are not critically needed” and ”Researchers who have found the opposite have typically argued either that viewers’ higher level of familiarity with cluttered charts make those charts more attractive or that decluttered charts that are too minimalistic become boring.”’ ‘28 Chartjunk/visual embellishments can have an impact on memory retention, with some studies suggesting that pictorial elements can improve memory for the fact that a visualization was previously seen. However, for better comprehension and understanding, a minimalist aesthetic is often recommended to avoid overwhelming the viewer with unnecessary design elements.’ ‘29 The data table is detectable and chartjunk/visual embellishments were not detected.’ ‘30 A bar chart is recommended for the data table as it allows for easy comparison of values across different categories and helps identify trends or relationships in the data.’ ‘31 The optimal chart for comparing values across different categories and identifying trends in the data table is a bar chart, while visual embellishments, such as chartjunk, were not detected but simplicity is recommended to improve memory retention.’ ‘32 The chart contains multiple colors.’ ‘33 ”The chart contains multiple colors listed.”’ ‘34 Colors can effectively represent information, but their effectiveness differ based on the type of data being visualized. Here are the rankings of colors in data visualization for different types of data:

1. Quantitative data: The most effective perceptual tasks for colors in visualizations of quantitative data are color saturation and color hue.
2. Ordinal data: For ordinal data, the most effective perceptual tasks for colors in visualizations are color saturation and color hue as well. However, density can also be a useful tool for representing ordinal data.
3. Nominal data: When visualizing nominal data, color hue is the most effective perceptual task for colors in visualizations. Density and color saturation can also be useful in representing nominal data effectively.

Overall, it is important to consider the type of data being visualized and choose colors that best represent the information in a clear and easily interpretable manner.’ ‘35 The variability of colors in data visualization depends

on the type of data being represented. For quantitative data, color saturation and hue are most effective. For ordinal data, saturation and hue are important, and density can also be useful. For nominal data, color hue is the most effective. Consider the type of data when selecting colors for visualization.’ ‘36 There are similar colors in the chart image.’ ‘37 There are similar colors in the chart image. ‘ When continuous values are encoded through visual channels, those values can be warped by categorical perception. One example is the rainbow. The same phenomenon occurs when data are depicted by hues.’ ‘38 Similar colors in visualizations can cause categorical perception and warp the interpretation of continuous data values.’ ‘39 The chart contains multiple colors, including similar colors, indicating variability within the color palette.’ ‘40 Choose colors based on the type of data being visualized, with attention to color saturation and hue for quantitative and ordinal data, and color hue for nominal data, while avoiding similar colors that can warp perception.’ ‘41 The effectiveness of colors in data visualization varies based on the type of data being visualized, with color saturation and hue being most effective for quantitative and ordinal data, and color hue being most effective for nominal data.’ ‘42 There is noticeable information loss in the chart image when converted to how people with Deuteranopia would see.’ ‘43 Use a colorblind-friendly palette when appropriate. For example, blue/orange is a common colorblind-friendly palette. Blue/red or blue/brown would also work. For the most common conditions of color-vision deficiencies, all of these work well, since blue would generally look blue to someone with color vision deficiencies. (taken from www.tableau.com).’ ‘44 Summary: The responses suggest that there is noticeable information loss in images for individuals with Deuteranopia, and recommend using a colorblind-friendly palette such as blue/orange, blue/red, or blue/brown to improve accessibility.’ ‘45 Answer: Use a colorblind-friendly palette in visuals to improve accessibility for individuals with Deuteranopia.’ ‘46 There is noticeable information loss in the chart image when converted to how people with Protanopia would see.’ ‘47 here is noticeable information loss in the chart image when converted to how people with Protanopia would see. Use a colorblind-friendly palette when appropriate. For example, blue/orange is a common colorblind-friendly palette. Blue/red or blue/brown would also work. For the most common conditions of color-vision deficiencies, all

of these work well, since blue would generally look blue to someone with color vision deficiencies. (taken from www.tableau.com).’ ‘48 Previous responses suggest that there is noticeable information loss in the chart image when converted to how people with protanopia would see. It is recommended to use a colorblind-friendly palette such as blue/orange, blue/red, or blue/brown to ensure information is accessible for individuals with color vision deficiencies. This approach can help improve the clarity and readability of data visualization for a wider audience.’ ‘49 There is noticeable information loss in the chart image when converted to how people with Tritanopia would see.’ ‘50 Use a colorblind-friendly palette when appropriate. For example, blue/orange is a common colorblind-friendly palette. Blue/red or blue/brown would also work. For the most common conditions of color-vision deficiencies, all of these work well, since blue would generally look blue to someone with color vision deficiencies. (taken from www.tableau.com).’ ‘51 Responses from past questions about Tritanopia: - Information loss in the chart image when converted to how people with Tritanopia would see. - Suggestion to use a colorblind-friendly palette, such as blue/orange, blue/red, or blue/brown. - Blue is a good color to use for color-vision deficiencies.’ ‘52 Color vision deficiency (Deutanopia, Protanopia, Tritanopia) results in noticeable information loss in how the affected individuals perceive colors.’ ‘53 Use a colorblind-friendly palette, such as blue/orange, blue/red, or blue/brown, to ensure that colors remain distinguishable for individuals with color vision deficiencies.’

A.2 Evaluation with ChatGPT and Wolfram-GPT

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
Experts	4.67	2.67	3.67	3.00	5.00	4.33	2.67	3.00	4.67	4.00	3.67	2.67	4.00
ChatGPT	4.53	4.40	4.00	4.20	4.40	4.07	4.07	4.13	4.47	4.53	4.40	4.00	4.07
WolframGPT	4.60	4.07	4.07	4.07	4.80	4.60	3.80	4.40	4.33	4.60	4.27	3.87	4.13

Table A.1: Comparison of averaged feedback scores of participants’ designs between 3 experts, 3 ChatGPT trials, and 3 WolframGPT trials.

To understand how large language models perceive visualization designs differently from

humans, I conducted an additional experiment with ChatGPT and WolframGPT to assess visualization improvements, similar to how three seasoned visualization experts evaluated 13 visualizations. Drawing from Ha et al. [207]’s work, I described the LLMs as visualization experts with more than five years of experience in data visualization, tasked with providing design feedback on visualizations.

I provided the preamble (Section A.1.1) but not the detailed prompt information as I did for Visualizationary. However, I asked for feedback on six topics (excluding visual saliency), identical to those that Visualizationary addresses.

For each iteration, I showed two images—one original and one final—and asked the LLMs to assess the improvements made between those two images. I repeated this three times for each participant’s image and for each language model. I provided both quantitative and qualitative comparisons between the experts’ comments and the two language models.

Quantitative comparison Table A.1 shows the averaged feedback scores of participants’ designs between 3 experts, 3 ChatGPT trials, and 3 WolframGPT trials. On average, ChatGPT and WolframGPT are more generous to evaluation. While experts’ scores on average was 3.69, ChatGPT’s and WolframGPT’s scores were 4.25, and 4.55, respectively. Also, experts’ standard deviation was 0.81, whereas ChatGPT’s and WolframGPT’s deviations were 0.21 and 0.31. This accounts to the fact that ChatGPT and WolframGPT tends to provide favorable feedback for users. One notable fact, however, was that the Pearson correlation between the experts’ scores and those from WolframGPT were 0.81. This shows that while WolframGPT’s scores were more generous than experts, the ability of WolframGPT in assessing visualization improvements shows strong correlation to how the experts evaluated the best, and the worst improvements.

Source	Feedback
An expert	Although context is added with a chart title, overall readability is reduced in terms of the scale of the visualization. There is more order to the data with many of the points placed along the top and left of the matrix though neither the X or Y axis is ordered, making it difficult to quickly find a point from either cluster.
ChatGPT	The second visualization shows an overall improvement in clarity, interpretability, and color usage. The context provided by the title and the better differentiation through color schemes help in understanding the data more intuitively. While the text could still be enhanced for further context, the improvements in visual representation are notable.
WolframGPT	The improvements made from the first to the last image are slight but noticeable, particularly in terms of color choice. The transition to a blue color scheme provides better differentiation and accessibility, while the overall representation and clarity have also seen positive changes. However, text presentation remains neutral with no significant improvement.

Table A.2: An example of design feedback about P7’s works from an expert, ChatGPT, and WolframGPT

Qualitative comparison From the explanations, both ChatGPT and WolframGPT are capable of detecting changes made in the chart image, such as color cues, sizes, and the increase of text. However, when assessing the changes, there are differences between how experts and LLMs do so. In general, both ChatGPT and WolframGPT tend to view the changes mostly positively. This is predictable due to the LLMs’ tendency to provide positive responses for users.

However, compared to experts, LLMs are not fully capable of assessing whether the changes positively or negatively affect the entire design. Although WolframGPT performs slightly better than ChatGPT when looking at quantitative figures (see Table A.1), neither can determine if the change is effective enough to convey the intended message of the designer. Table A.2 shows an example of textual feedback provided by an expert, ChatGPT, and WolframGPT. While the expert evaluates the impact of the change based on how it helps understand the data, both ChatGPT and WolframGPT are not capable of describing the visualization on a detailed data level. This

indicates that while vision-language models can detect changes in the image, they do not have human-like perception. Therefore, such perception must be guided to provide helpful feedback.

Bibliography

- [1] Laura E. Matzen, Michael J. Haass, Kristin M. Divis, Zhiyuan Wang, and Andrew T. Wilson. Data visualization saliency model: A tool for evaluating abstract data visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):563–573, 2018.
- [2] M. Jiang, S. Huang, J. Duan, and Q. Zhao. SALICON: Saliency in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1072–1080, Piscataway, NJ, USA, 2015. IEEE.
- [3] Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediya Daskalova, Jeff Huang, and James Hays. Webgazer: Scalable webcam eye tracking using user interactions. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3839–3845. AAAI Press, 2016.
- [4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, Cham, 2015. Springer International Publishing.
- [5] Scott Bateman, Regan L. Mandryk, Carl Gutwin, Aaron Genest, David McDine, and Christopher A. Brooks. Useful junk?: the effects of visual embellishment on comprehension and memorability of charts. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 2573–2582, New York, NY, USA, 2010. ACM.
- [6] Sungbok Shin, Sunghyo Chung, Sanghyun Hong, and Niklas Elmqvist. A scanner deeply: Predicting gaze heatmaps on visualizations using crowdsourced eye movement data. *IEEE Transactions on Visualization and Computer Graphics*, 29(1), 2023.
- [7] Sungbok Shin, Sanghyun Hong, and Niklas Elmqvist. Perceptual pat: A virtual human visual system for iterative visualization design. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, New York, NY, USA, 2023. Association for Computing Machinery.
- [8] C. Stolte, D. Tang, and P. Hanrahan. Polaris: a system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002.
- [9] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

- [10] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, 2017.
- [11] Donghao Ren, Bongshin Lee, and Matthew Brehmer. Charticulator: Interactive construction of bespoke chart layouts. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):789–799, 2019.
- [12] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, 2016.
- [13] Dominik Moritz, Chenglong Wang, Greg L. Nelson, Halden Lin, Adam M. Smith, Bill Howe, and Jeffrey Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):438–448, 2019.
- [14] Halden Lin, Dominik Moritz, and Jeffrey Heer. Dziban: Balancing agency & automation in visualization design via anchored recommendations. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, page 1–12, New York, NY, USA, 2020. ACM.
- [15] Kevin Hu, Michiel A. Bakker, Stephen Li, Tim Kraska, and César Hidalgo. VizML: A machine learning approach to visualization recommendation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, page 128:1–128:12, New York, NY, USA, 2019. ACM.
- [16] Andrew McNutt, Gordon Kindlmann, and Michael Correll. Surfacing visualization mirages. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, page 1–16, New York, NY, USA, 2020. ACM.
- [17] Andrew McNutt and Gordon Kindlmann. Linting for visualization: Towards a practical automated visualization guidance system. In *Proceedings of the VisGuides: 2nd Workshop on the Creation, Curation, Critique and Conditioning of Principles and Guidelines in Visualization*, pages 1–14, 2018.
- [18] Aspen K. Hopkins, Michael Correll, and Arvind Satyanarayan. VisuaLint: Sketchy in situ annotations of chart construction errors. *Computer Graphics Forum*, 39(3):219–228, 2020.
- [19] Qing Chen, Fuling Sun, Xinyue Xu, Zui Chen, Jiazhe Wang, and Nan Cao. VizLinter: A linter and fixer framework for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):206–216, 2022.
- [20] Mengchen Liu, Jiaxin Shi, Zhen Li, Chongxuan Li, Jun Zhu, and Shixia Liu. Towards better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):91–100, 2017.

- [21] Mengchen Liu, Jiaxin Shi, Kelei Cao, Jun Zhu, and Shixia Liu. Analyzing the training processes of deep generative models. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):77–87, 2018.
- [22] Minsuk Kahng, Nikhil Thorat, Duen Horng Chau, Fernanda B. Viégas, and Martin Wattenberg. Gan lab: Understanding complex deep generative models using interactive visual experimentation. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):310–320, 2019.
- [23] Y. Ming, H. Qu, and E. Bertini. Rulematrix: Visualizing and understanding classifiers with rules. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):342–352, 2019.
- [24] Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M. Drucker. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery.
- [25] F. Hohman, H. Park, C. Robinson, and D. H. Polo Chau. Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1096–1106, 2020.
- [26] Zijie J. Wang, Robert Turko, Omar Shaikh, Haekyu Park, Nilaksh Das, Fred Hohman, Minsuk Kahng, and Duen Horng Polo Chau. Cnn explainer: Learning convolutional neural networks with interactive visualization. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1396–1406, 2021.
- [27] Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, 2013.
- [28] Leilani Battle, Peitong Duan, Zachery Miranda, Dana Mukusheva, Remco Chang, and Michael Stonebraker. Beagle: Automated extraction and interpretation of visualizations from the web. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, page 1–8, New York, NY, USA, 2018. Association for Computing Machinery.
- [29] Robert Pienta, Fred Hohman, Alex Endert, Acar Tamersoy, Kevin Roundy, Chris Gates, Shamkant Navathe, and Duen Horng Chau. Vigor: Interactive visual exploration of graph query results. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):215–225, 2018.
- [30] Stephen Smart, Keke Wu, and Danielle Albers Szafir. Color crafting: Automating the construction of designer quality color ramps. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1215–1225, 2020.
- [31] Qianwen Wang, Zhutian Chen, Yong Wang, and Huamin Qu. A survey on ml4vis: Applying machine learning advances to data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):5134–5153, 2022.

- [32] Ben Shneiderman. *Human-centered AI*. Oxford University Press, 2022.
- [33] R. Shi, N. K. Ngan, and H. Li. Gaze-based object segmentation. *IEEE Signal Processing Letters*, 24(10):1493–1497, 2017.
- [34] S. Karthikeyan, Thuyen Ngo, M. Eckstein, and B. S. Manjunath. Eye tracking assisted extraction of attentionally important objects from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3241–3250, Piscataway, NJ, USA, 2015. IEEE.
- [35] Robert J.K. Jacob and Keith S. Karn. Commentary on section 4 - eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In J. Hyona, R. Radach, and H. Deubel, editors, *The Mind’s Eye*, pages 573–605. North-Holland, Amsterdam, 2003.
- [36] Päivi Majaranta and Andreas Bulling. *Eye Tracking and Eye-Based Human–Computer Interaction*, pages 39–65. Springer London, London, 2014.
- [37] Kathy Conklin and Ana Pellicer-Sánchez. Using eye-tracking in applied linguistics and second language research. *Second Language Research*, 32(3):453–467, 2016.
- [38] Corinna E. Bonhage, Jutta L. Mueller, Angela D. Friederici, and Christian J. Fiebach. Combined eye tracking and fMRI reveals neural basis of linguistic predictions during sentence comprehension. *Cortex*, 68:33–47, 2015.
- [39] Sung-Hee Kim, Zhihua Dong, Hanjun Xian, Benjavan Upatising, and Ji Soo Yi. Does an eye tracker tell the truth about visualizations?: Findings while investigating visualizations for decision making. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2421–2430, 2012.
- [40] Michelle A. Borkin, Zoya Bylinskii, Nam Wook Kim, Constance May Bainbridge, Chelsea S. Yeh, Daniel Borkin, Hanspeter Pfister, and Aude Oliva. Beyond memorability: Visualization recognition and recall. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):519–528, 2016.
- [41] Nam Wook Kim, Zoya Bylinskii, Michelle A. Borkin, Krzysztof Z. Gajos, Aude Oliva, Fredo Durand, and Hanspeter Pfister. BubbleView: An interface for crowdsourcing image importance maps and tracking visual attention. *ACM Transactions on Computer-Human Interaction*, 24(5), 2017.
- [42] Anelise Newman, Barry McNamara, Camilo Fosco, Yun Bin Zhang, Pat Sukhum, Matthew Tancik, Nam Wook Kim, and Zoya Bylinskii. TurkEyes: A web-based toolbox for crowdsourcing attention data. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, page 1–13, New York, NY, USA, 2020. ACM.
- [43] Navyasri Reddy, Samyak Jain, Pradeep Yarlagadda, and Vineet Gandhi. Tidying deep saliency prediction architectures. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 10241–10247, Piscataway, NJ, USA, 2020. IEEE.

- [44] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
- [45] Weidong Huang. Using eye tracking to investigate graph layout effects. In *Proceedings of the International Asia-Pacific Symposium on Visualization*, pages 97–100, Feb 2007.
- [46] H. Siirtola, T. Laivo, T. Heimonen, and K. Räihä. Visual perception of parallel coordinate visualizations. In *Proceedings of the IEEE International Conference on Information Visualisation*, pages 3–9, Piscataway, NJ, USA, July 2009. IEEE.
- [47] T. Judd, K. Ehinger, F. Durand, and A. Torralba. Learning to predict where humans look. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2106–2113, Piscataway, NJ, USA, 2009. IEEE.
- [48] Erik D. Reichle, Andrew E. Reineberg, and Jonathan W. Schooler. Eye movements during mindless reading. *Psychological Science*, 21(9):1300–1310, 2010.
- [49] M. Burch, N. Konevtsova, J. Heinrich, M. Hoferlin, and D. Weiskopf. Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2440–2448, 2011.
- [50] Ben Steichen, Giuseppe Carenini, and Cristina Conati. User-adaptive information visualization: Using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proceedings of the ACM Conference on Intelligent User Interfaces*, page 317–328, New York, NY, USA, 2013. ACM.
- [51] Natasha Jaques, Cristina Conati, Jason M. Harley, and Roger Azevedo. Predicting affect from gaze data during interaction with an intelligent tutoring system. In Stefan Trausan-Matu, Kristy Elizabeth Boyer, Martha Crosby, and Kitty Panourgia, editors, *Intelligent Tutoring Systems*, pages 29–38, Cham, 2014. Springer International Publishing.
- [52] Nam Wook Kim, Zoya Bylinskii, Michelle A. Borkin, Aude Oliva, Krzysztof Z. Gajos, and Hanspeter Pfister. A crowdsourced alternative to eye-tracking for visualization understanding. In *Proceedings of the ACM Conference Extended Abstracts on Human Factors in Computing Systems*, page 1349–1354, New York, NY, USA, 2015. ACM.
- [53] Pingmei Xu, Krista A. Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R. Kulkarni, and Jianxiong Xiao. TurkerGaze: Crowdsourcing saliency with webcam based eye tracking. *CoRR*, abs/1504.06755, 2015.
- [54] Matthew Junghyun Gingerich and Cristina Conati. Constructing models of user and task characteristics from eye gaze data for user-adaptive information highlighting. In *Proceedings of the AAAI conference on artificial intelligence*, 2015.
- [55] Sébastien Lallé, Cristina Conati, and Giuseppe Carenini. Predicting confusion in information visualization from eye tracking and interaction data. In *Proceedings of the International Joint Conference on Artificial Intelligence*, page 2529–2535, New York, New York, USA, 2016. AAAI Press.

- [56] Benjamin Strobel, Steffani Sass, Marlit Annalena Lindner, and Olaf Köller. Do graph readers prefer the graph type most suited to a given task? insights from eye tracking. *Journal of Eye Movement Research*, 9(4):1–15, 2016.
- [57] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. Eye tracking for everyone. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2176–2184, 2016.
- [58] Sidney K D’Mello, Caitlin Mills, Robert Bixler, and Nigel Bosch. Zone out no more: Mitigating mind wandering during computerized reading. *International Educational Data Mining Society*, 2017.
- [59] Zoya Bylinskii, Nam Wook Kim, Peter O’Donovan, Sami Alsheikh, Spandan Madan, Hanspeter Pfister, Fredo Durand, Bryan Russell, and Aaron Hertzmann. Learning visual importance for graphic designs and data visualizations. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, page 57–69, New York, NY, USA, 2017. ACM.
- [60] Alexandra Papoutsaki, James Laskey, and Jeff Huang. Searchgazer: Webcam eye tracking for remote studies of web search. In *Proceedings of the ACM Conference on Human Information Interaction and Retrieval*, page 17–26, New York, NY, USA, 2017. ACM.
- [61] Steven R. Gomez, Radu Jianu, Ryan Cabeen, Hua Guo, and David H. Laidlaw. Fauxvea: Crowdsourcing gaze location estimates for visualization analysis tasks. *IEEE Transactions on Visualization and Computer Graphics*, 23(2):1042–1055, 2017.
- [62] Hamed R. Tavakoli, Fawad Ahmed, Ali Borji, and Jorma Laaksonen. Saliency revisited: Analysis of mouse movements versus fixations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6354–6362, 2017.
- [63] Laura E. Matzen, Michael J. Haass, Kristin M. Divis, and Mallory C. Stites. Patterns of attention: How data visualizations are read. In Dylan D. Schmorrow and Cali M. Fidopiastis, editors, *Augmented Cognition. Neurocognition and Machine Learning*, pages 176–191, Cham, 2017. Springer International Publishing.
- [64] Wenguan Wang, Jianbing Shen, Fang Guo, Ming-Ming Cheng, and Ali Borji. Revisiting video saliency: A large-scale benchmark and a new model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4894–4903, 2018.
- [65] Patrik Polatsek, Manuela Waldner, Ivan Viola, Peter Kapec, and Wanda Benesova. Exploring visual attention and saliency modeling for task-based visual analysis. *Computers & Graphics*, 72:26–38, 2018.
- [66] Camilo Fosco, Vincent Casser, Amish Kumar Bedi, Peter O’Donovan, Aaron Hertzmann, and Zoya Bylinskii. Predicting visual importance across graphic design types. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, page 249–260, New York, NY, USA, 2020. ACM.

- [67] Shiwei Cheng, Jing Fan, and Yilin Hu. Visual saliency model based on crowdsourcing eye tracking data and its application in visual design. *Personal and Ubiquitous Computing*, pages 1–18, 2020.
- [68] Sébastien Lalle, Tiffany Wu, and Cristina Conati. Gaze-driven links for magazine style narrative visualizations. In *Proceedings of the IEEE Visualization Conference*, pages 166–170, 2020.
- [69] Sukwon Lee, Sung-Hee Kim, Ya-Hsin Hung, Heidi Lam, Youn-Ah Kang, and Ji Soo Yi. How do people make sense of unfamiliar visualizations?: A grounded model of novice’s information visualization sensemaking. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):499–508, 2016.
- [70] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 19, Red Hook, NY, USA, 2006. Curran Associates, Inc.
- [71] Neil Bruce and John Tsotsos. Saliency based on information maximization. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 18, Red Hook, NY, USA, 2005. Curran Associates, Inc.
- [72] F. Lu, Y. Sugano, T. Okabe, and Y. Sato. Inferring human gaze from appearance via adaptive linear regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 153–160, Piscataway, NJ, USA, 2011. IEEE.
- [73] J. Pan, E. Sayrol, X. Giro-I-Nieto, K. McGuinness, and N. E. O’Connor. Shallow and deep convolutional networks for saliency prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 598–606, Piscataway, NJ, USA, 2016. IEEE.
- [74] Zoya Bylinskii, Tilke Judd, Ali Borji, Laurent Itti, Frédo Durand, Aude Oliva, and Antonio Torralba. MIT saliency benchmark. <http://saliency.mit.edu/>, 2012.
- [75] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755, Cham, 2014. Springer International Publishing.
- [76] Matthias Kümmerer, Lucas Theis, and Matthias Bethge. Deep Gaze I: Boosting saliency prediction with feature maps trained on ImageNet. In Yoshua Bengio and Yann LeCun, editors, *Proceedings of the International Conference on Learning Representations*, 2015.
- [77] Matthias Kümmerer, , T. S. A. Wallis, L. A. Gatys, and M. Bethge. Understanding low- and high-level contributions to fixation prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4799–4808, Piscataway, NJ, USA, 2017. IEEE.
- [78] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. SAM: Pushing the limits of saliency prediction models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1971–19712, Piscataway, NJ, USA, 2018. IEEE.

- [79] Michael J Haass, Andrew T Wilson, Laura E Matzen, and Kristin M Divis. Modeling human comprehension of data visualizations. In *Proceedings of the International Conference on Virtual, Augmented and Mixed Reality*, pages 125–134. Springer, 2016.
- [80] Jeff Huang, Ryen White, and Georg Buscher. User see, user point: Gaze and cursor alignment in web search. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 1341–1350, New York, NY, USA, 2012. ACM.
- [81] Zoya Bylinskii, Michelle A. Borkin, Nam Wook Kim, Hanspeter Pfister, and Aude Oliva. Eye fixation metrics for large scale evaluation and comparison of information visualizations. In Michael Burch, Lewis Chuang, Brian Fisher, Albrecht Schmidt, and Daniel Weiskopf, editors, *Eye Tracking and Visualization*, pages 235–255. Springer International Publishing, 2017.
- [82] Alfred L Yarbus. Eye movements during perception of complex objects. In *Eye Movements and Vision*, pages 171–211. Springer, Boston, MA, USA, 1967.
- [83] Audrey L. Michal and Steven L. Franconeri. Visual routines are associated with specific graph interpretations. *Cognitive Research: Principles and Implications*, 2(1):1–10, 2017.
- [84] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, Piscataway, NJ, USA, 2016. IEEE.
- [85] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [86] Eleonora Vig, Michael Dorr, and David Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2798–2805, 2014.
- [87] Akis Linardos, Matthias Kümmerer, Ori Press, and Matthias Bethge. Deepgaze iie: Calibrated prediction in and out-of-domain for state-of-the-art saliency modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12919–12928, October 2021.
- [88] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015.
- [89] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Proceedings of the European Conference on Computer Vision*, pages 19–35, Cham, 2018. Springer International Publishing.
- [90] Laura Renninger, James Coughlan, Preeti Verghese, and Jitendra Malik. An information maximization model of eye movements. In L. Saul, Y. Weiss, and L. Bottou, editors,

Proceedings of the Advances in Neural Information Processing Systems, volume 17. MIT Press, 2004.

- [91] T. Kadir and M. Brady. Scale saliency: a novel approach to salient feature and scale selection. In *Proceedings of the International Conference on Visual Information Engineering*, pages 25–28, 2003.
- [92] Goldberger, Gordon, and Greenspan. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 487–493 vol.1, 2003.
- [93] Olivier Le Meur and Thierry Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods*, 45(1):251–266, 2013.
- [94] Nicolas Riche, Matthieu Duvinage, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. Saliency and human fixations: State-of-the-art and study of comparison metrics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1153–1160, Piscataway, NJ, USA, 2013. IEEE.
- [95] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [96] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(3):740–757, 2019.
- [97] Taralee Hamner and Giacomo Vivanti. Eye-tracking research in autism spectrum disorder: What are we measuring and for what purposes? *Current Developmental Disorders Reports*, 6(2):37–44, 2019.
- [98] Daniel Haehn, James Tompkin, and Hanspeter Pfister. Evaluating “graphical perception” with CNNs. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):641–650, 2018.
- [99] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, volume 139, pages 8748–8763. PMLR, 2021.
- [100] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

- [101] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, Boca Raton, FL, USA, 2014.
- [102] Norman I. Badler, Cary B. Phillips, and Bonnie Lynn Webber. *Simulating Humans: Computer Graphics, Animation, and Control*. Oxford University Press, Oxford, United Kingdom, 09 1993.
- [103] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1983.
- [104] Tamara Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, November 2009.
- [105] Michael Sedlmair, Miriah D. Meyer, and Tamara Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.
- [106] Jeffrey Bardzell, Jay D. Bolter, and Jonas Löwgren. Interaction criticism: three readings of an interaction design, and what they get us. *Interactions*, 17(2):32–37, 2010.
- [107] Jeffrey Bardzell. Interaction criticism: An introduction to the practice. *Interacting with Computers*, 23(6):604–621, 2011.
- [108] Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, 1986.
- [109] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, 2007.
- [110] Alicia Key, Bill Howe, Daniel Perry, and Cecilia Aragon. VizDeck: Self-organizing dashboards for visual analytics. In *Proceedings of the ACM Conference on Management of Data*, page 681–684, New York, NY, USA, 2012. ACM.
- [111] Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. DeepEye: Towards automatic data visualization. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 101–112, Piscataway, NJ, USA, 2018. IEEE.
- [112] Arlen Fan, Yuxin Ma, Michelle Mancenido, and Ross Maciejewski. Annotating line charts for addressing deception. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 80:1–80:12, New York, NY, USA, 2022. ACM.
- [113] Ben Shneiderman. *Human-Centered AI*. Oxford University Press, Oxford, United Kingdom, 2022.
- [114] Kurt Koffka. Perception: An introduction to the Gestalt-theorie. *Psychological Bulletin*, 19:531–585, 1922.
- [115] Walter C. Eells. The relative merits of circles and bars for representing component parts. *Journal of the American Statistical Association*, 21(154):119–132, 1926.

- [116] Frederick E. Croxton and Roy E. Stryker. Bar charts versus circle diagrams. *Journal of the American Statistical Association*, 22(160):473–482, 1927.
- [117] Frederick E. Croxton and Harold Stein. Graphic comparisons by bars, squares, circles, and cubes. *Journal of the American Statistical Association*, 27(177):54–60, 1932.
- [118] Lewis V. Peterson and Wilbur Schramm. How accurately are different kinds of graphs read? *Educational Technology Research and Development*, 2(3):178–189, June 1954.
- [119] William S. Cleveland and Robert McGill. Graphical perception: Theory, experimentation and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.
- [120] Amira Chalbi, Jacob Ritchie, Deokgun Park, Jungu Choi, Nicolas Roussel, Niklas Elmqvist, and Fanny Chevalier. Common fate for animated transitions in visualization. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):386–396, 2020.
- [121] Steven L. Franconeri, Lace M. Padilla, Priti Shah, Jeffrey M. Zacks, and Jessica Hullman. The science of visual data communication: What works. *Psychological Science in the Public Interest*, 22(3):110–161, 2021.
- [122] Dereck Toker, Ben Steichen, Matthew Gingerich, Cristina Conati, and Giuseppe Carenini. Towards facilitating user skill acquisition: Identifying untrained visualization users through eye tracking. In *Proceedings of the International Conference on Intelligent User Interfaces*, page 105–114, New York, NY, USA, 2014. ACM.
- [123] Dereck Toker, Cristina Conati, Ben Steichen, and Giuseppe Carenini. Individual user characteristics and information visualization: Connecting the dots through eye tracking. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, page 295–304, New York, NY, USA, 2013. ACM.
- [124] Gerald L. Lohse. A cognitive model for understanding graphical perception. *Human-Computer Interaction*, 8(4):353–388, 1993.
- [125] Daniel Haehn, James Tompkin, and Hanspeter Pfister. Evaluating ‘graphical perception’ with cnns. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):641–650, 2019.
- [126] Catherine Plaisant. The challenge of information visualization evaluation. In *Proceedings of the ACM Conference on Advanced Visual Interfaces*, pages 109–116, New York, NY, USA, 2004. ACM.
- [127] Tobias Isenberg, Petra Isenberg, Jian Chen, Michael Sedlmair, and Torsten Möller. A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2818–2827, 2013.
- [128] Michael Sedlmair, Petra Isenberg, Dominikus Baur, and Andreas Butz. Information visualization evaluation in large companies: Challenges, experiences and recommendations. *Information Visualization*, 10(3):248–266, 2011.

- [129] Min Chen and Heike Jänicke. An information-theoretic framework for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1206–1215, 2010.
- [130] Theresa-Marie Rhyne. *Applying Color Theory to Digital Media and Visualization*. CRC Press, Boca Raton, FL, USA, 2016.
- [131] Christopher G. Healey. Choosing effective colours for data visualization. In *Proceedings of the IEEE Visualization Conference*, pages 263–270, Piscataway, NJ, USA, 1996. IEEE.
- [132] Danielle Albers Szafir. Modeling color difference for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):392–401, 2018.
- [133] Ewald Hering. *Outlines of a Theory of the Light Sense*. Harvard University Press, Cambridge, MA, USA, 1964.
- [134] Leo M. Hurvich and Dorothea Jameson. An opponent-process theory of color vision. *Psychological Review*, 64(6, Pt.1):384–404, 1957.
- [135] Jinho Choi, Sanghun Jung, Deok Gun Park, Jaegul Choo, and Niklas Elmqvist. Visualizing for the non-visual: Enabling the visually impaired to use visualization. *Computer Graphics Forum*, 38(3):249–260, 2019.
- [136] Jorge Poco and Jeffrey Heer. Reverse-engineering visualizations: Recovering visual encodings from chart images. *Computer Graphics Forum*, 36(3):353–363, June 2017.
- [137] Manolis Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and Jeffrey Heer. ReVision: Automated classification, analysis and redesign of chart images. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 393–402, New York, NY, USA, 2011. ACM.
- [138] William S. Cleveland. *The Elements of Graphing Data*. Wadsworth, Monterey, CA, USA, 1985.
- [139] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.
- [140] Anne Treisman. Preattentive processing in vision. *Computer Vision, Graphics and Image Processing*, 31:156–177, 1985.
- [141] Jeremy M. Wolfe and Igor S. Utochkin. What is a preattentive feature? *Current Opinion in Psychology*, 29:19–26, 2019.
- [142] George A. Alvarez. Representing multiple objects as an ensemble enhances visual cognition. *Trends in Cognitive Sciences*, 15(3):122–131, 2011.
- [143] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542, 2022.

- [144] Linda G. Shapiro and George C. Stockman. *Computer Vision*. Prentice-Hall, Hoboken, NJ, USA, 2001.
- [145] Patrick Cavanagh and George A. Alvarez. Tracking multiple targets with multifocal attention. *Trends in Cognitive Sciences*, 9(7):349–354, 2005.
- [146] Zenon W. Pylyshyn and R. W. Storm. Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision*, 3:179–197, 1988.
- [147] Pierre Dragicevic, Anastasia Bezerianos, Waqas Javed, Niklas Elmqvist, and Jean-Daniel Fekete. Temporal distortion for animated transitions. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 2009–2018, New York, NY, USA, 2011. ACM.
- [148] Pramod Chundury, Biswaksen Patnaik, Yasmin Reyazuddin, Christine Tang, Jonathan Lazar, and Niklas Elmqvist. Towards understanding sensory substitution for accessible visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1084–1094, 2022.
- [149] Katrin Angerbauer, Nils Rodrigues, René Cutura, Seyda Öney, Nelusa Pathmanathan, Cristina Morariu, Daniel Weiskopf, and Michael Sedlmair. Accessibility for color vision deficiencies: Challenges and findings of a large scale study on paper figures. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 134:1–134:23, New York, NY, USA, 2022. ACM.
- [150] Frank Elavsky, Cynthia L. Bennett, and Dominik Moritz. How accessible is my visualization? evaluating visualization accessibility with chartability. *Computer Graphics Forum*, 41(3):57–70, 2022.
- [151] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, and Jake VanderPlas. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, 2020.
- [152] Anthony Kay. Tesseract: An open-source optical character recognition engine. *Linux J.*, 2007(159):2, 2007.
- [153] Tesseract-OCR. Performance of tessarct-ocr v.4.0. <https://github.com/tesseract-ocr/tessdoc/blob/main/tess4/4.0-Accuracy-and-Performance.md>, 1999.
- [154] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks. *CoRR*, abs/2105.04206, 2021.
- [155] Lamiaa A. Elrefaei. Smartphone based image color correction for color blindness. *International Journal of Interactive Mobile Technologies*, 12(3):104–119, 2018.

- [156] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2017.
- [157] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755, Cham, 2014. Springer International Publishing.
- [158] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Performance of yolor. <https://github.com/WongKinYiu/yolor>, 2021.
- [159] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, Piscataway, NJ, USA, 2017. IEEE.
- [160] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *Proceedings of the International Conference on Machine Learning*, pages 3301–3310, Long Beach, CA, United States, 2019. PMLR.
- [161] Asma Ghandeharioun, Been Kim, Chun-Liang Li, Brendan Jou, Brian Eoff, and Rosalind W. Picard. DISSECT: disentangled simultaneous explanations via concept traversals, 2021.
- [162] Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. *CoRR*, abs/2011.05429, 2020.
- [163] Donghao Ren, Tobias Höllerer, and Xiaoru Yuan. iVisDesigner: Expressive interactive design of information visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2092–2101, 2014.
- [164] Zhicheng Liu, John Thompson, Alan Wilson, Mira Dontcheva, James Delorey, Sam Grigg, Bernard Kerr, John Stasko, and UT Lehi. Data Illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 123:1–123:13, 2018.
- [165] John R. Thompson, Zhicheng Liu, and John T. Stasko. Data animator: Authoring expressive animated data graphics. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 15:1–15:18, New York, NY, USA, 2021. ACM.
- [166] Donghao Ren, Bongshin Lee, and Matthew Brehmer. Charticulator: Interactive construction of bespoke chart layouts. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):789–799, 2019.
- [167] Arvind Satyanarayan and Jeffrey Heer. Lyra: An interactive visualization design environment. *Computer Graphics Forum*, 33(3):351–360, 2014.

- [168] Jonathan Zong, Dhiraj Barnwal, Rupayan Neogy, and Arvind Satyanarayan. Lyra 2: Designing interactive visualizations by demonstration. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):304–314, 2021.
- [169] Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhua Chen, Nigel Collier, and Yasemin Altun. DePlot: One-shot visual language reasoning by plot-to-table translation. In *Proceedings of the Findings of the Association for Computational Linguistics*, pages 10381–10399, Toronto, Canada, 2023. Association for Computational Linguistics.
- [170] Stephen Few. *Information Dashboard Design: The Effective Visual Communication of Data*. O’Reilly Press, 2006.
- [171] Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock D. Mackinlay, Bill Howe, and Jeffrey Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 2648–2659, New York, NY, USA, 2017. ACM.
- [172] Niklas Elmquist. Visualization for the blind. *ACM Interactions*, 30(1):52–56, 2023.
- [173] Kim Marriott, Bongshin Lee, Matthew Butler, Ed Cutrell, Kirsten Ellis, Cagatay Goncu, Marti A. Hearst, Kathleen F. McCoy, and Danielle Albers Szafr. Inclusive data visualization for people with disabilities: a call to action. *Interactions*, 28(3):47–51, 2021.
- [174] Jinhan Choi, Changhoon Oh, Yea-Seul Kim, and Nam Wook Kim. VisLab: Enabling visualization designers to gather empirically informed design feedback. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 813:1–813:18, New York, NY, USA, 2023. ACM.
- [175] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, Red Hook, NY, USA, 2020. Curran Associates, Inc.
- [176] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *CoRR*, abs/2206.07682, 2022.
- [177] Deep Ganguli and et al. Predictability and surprise in large generative models. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, pages 1747–1764, New York, NY, USA, 2022. ACM.
- [178] OpenAI. ChatGPT, 2023.
- [179] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality, March 2023.

- [180] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023.
- [181] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernández Ábrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, and et al. PaLM 2 technical report. *CoRR*, abs/2305.10403, 2023.
- [182] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling language modeling with pathways. *CoRR*, abs/2204.02311, 2022.
- [183] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d’Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew J. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Edward Lockhart, Simon Osindero, and

- et al. Scaling language models: Methods, analysis & insights from training Gopher. *CoRR*, abs/2112.11446, 2021.
- [184] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel J. Orr, Lucia Zheng, Mert Yüsekçönül, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models. *CoRR*, abs/2211.09110, 2022.
 - [185] Tanya Goyal, Junyi Jessy Li, and Greg Durrett. News summarization and evaluation in the era of GPT-3. *CoRR*, abs/2209.12356, 2022.
 - [186] Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhua Chen, Nigel Collier, and Yasemin Altun. DePlot: One-shot visual language reasoning by plot-to-table translation. *CoRR*, abs/2212.10505, 2022.
 - [187] Jianyu Wang, Anit Kumar Sahu, Zhouyi Yang, Gauri Joshi, and Soumya Kar. MATCHA: speeding up decentralized SGD via matching decomposition sampling. *CoRR*, abs/1905.09435, 2019.
 - [188] Chase Stokes, Vidya Setlur, Bridget Cogley, Arvind Satyanarayan, and Marti A. Hearst. Striking a balance: Reader takeaways and preferences when integrating text and charts. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1233–1243, Jan 2023.
 - [189] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister. What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2306–2315, 2013.
 - [190] Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *Proceedings of the ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 1–7, New York, NY, USA, 2021. ACM.
 - [191] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, Red Hook, NY, USA, 2022. Curran Associates, Inc.
 - [192] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. *CoRR*, abs/2205.10625, 2022.

- [193] Simran Arora, Avanika Narayan, Mayee F. Chen, Laurel J. Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. Ask me anything: A simple strategy for prompting language models. *CoRR*, abs/2210.02441, 2022.
- [194] Vivian Liu and Lydia B. Chilton. Design guidelines for prompt engineering text-to-image generative models. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 1–23, New York, NY, USA, 2022. ACM.
- [195] Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, 1986.
- [196] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.
- [197] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.
- [198] Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. Factuality enhanced language models for open-ended text generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 34586–34599, Red Hook, NY, USA, 2022. Curran Associates, Inc.
- [199] Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin T. Vechev. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *CoRR*, abs/2305.15852, 2023.
- [200] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. *CoRR*, abs/2303.08896, 2023.
- [201] Mark Gurman. Samsung Bans Staff’s AI Use After Spotting ChatGPT Data Leak, 2023.
- [202] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the ACM Conference on Computer and Communications Security*, pages 619–633, New York, NY, USA, 2018. ACM.
- [203] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 1897–1914, Los Alamitos, CA, USA, 2022. IEEE Computer Society.
- [204] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *Proceedings of the USENIX Security Symposium*, pages 267–284, Berkeley, CA, USA, 2019. USENIX Association.

- [205] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *Proceedings of the USENIX Security Symposium*, pages 2633–2650, Berkeley, CA, USA, 2021. USENIX Association.
- [206] Can Liu, Yuhan Guo, and Xiaoru Yuan. Autotitle: An interactive title generator for visualizations. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–12, 2023.
- [207] Juhye Ha, Hyeon Jeon, Daeun Han, Jinwook Seo, and Changhoon Oh. Clochat: Understanding how people customize, interact, and experience personas in large language models. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, New York, NY, USA, 2024. Association for Computing Machinery.