Visualizationary: Automating Design Feedback for Visualization Designers Using LLMs

Sungbok Shin, Sanghyun Hong, and Niklas Elmqvist, *Fellow, IEEE*

Abstract—Interactive visualization editors empower users to author visualizations without writing code, but do not provide guidance on the art and craft of effective visual communication. In this paper, we explore the potential of using an off-the-shelf large language models (LLMs) to provide actionable and customized feedback to visualization designers. Our implementation, VISUALIZATIONARY, demonstrates how ChatGPT can be used for this purpose through two key components: a preamble of visualization design guidelines and a suite of perceptual filters that extract salient metrics from a visualization image. We present findings from a longitudinal user study involving 13 visualization designers—6 novices, 4 intermediates, and 3 experts—who authored a new visualization from scratch over several days. Our results indicate that providing guidance in natural language via an LLM can aid even seasoned designers in refining their visualizations. All our supplemental materials are available at https://osf.io/v7hu8.

Index Terms—Visualization design, Design critique, Feedback, Human-centered AI, large language models.

1 INTRODUCTION

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. In response, recent years have seen an influx of interactive tools that enable users to create visualizations without writing code, such as iVisDesigner [51], Data Illustrator [38], and Lyra [56]. However, no matter how advanced these tools become, a fundamental barrier remains: authoring effective visualizations is a complex task requiring visual design skills, an understanding of aesthetics, and experience—expertise that not all would-be designers possess.

Fortunately, visualization designers have nothing to lose but their chains. We introduce VISUALIZATIONARY, a system to seize the means of visualization design by using off-the-shelf large-language models (LLMs). Our system enhances the design of communicative visualizations by providing perceptual feedback. While it may not fully replace feedback from peers or colleagues, it serves as a valuable alternative when it is impossible to gain human feedback and can benefit designers of all expertise levels, from novices to seasoned ones.

Although competing approaches for improving visualization designs, such as heuristics (like small, smart defaults in Lyra) and Grammar of Graphics (GoG)-style specifications have their own advantages, they also carry limitations due to their deterministic nature. Heuristic-driven methods lack human intervention, which can constrain creativity and limit the potential for novel ideas. GoG-style specifications, despite offering precise chart descriptions, lead to producing standardized solutions, reducing the diversity of possible visualization outcomes.

- *Sungbok Shin is with Aviz at Inria and Université Paris-Saclay, Saclay, France. The work was done while the author was affiliated with the University of Maryland, College Park. E-mail: sungbok.shin@inria.fr*
- *Sanghyun Hong is with the Department of Computer Science at Oregon State University, Corvallis, OR, USA. E-mail: sanghyun.hong@oregonstate.edu*
- *Niklas Elmqvist is with the Department of Computer Science at Aarhus University, Aarhus, Denmark. E-mail: elm@cs.au.dk*

Manuscript received XXX XX, 2024; revised XXX XX, 2024.

In contrast, Visualizationary automates perceptual feedback using a suite of automated techniques that simulate how a viewer would perceive a visualization. By providing actionable insights without the need to standardize changes, Visualizationary preserves the designer's creativity while enhancing effectiveness.

The basic idea behind Visualizationary is to investigate how 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) **Analyze** the state of a visualization using automated perceptual filters, yielding a corresponding analysis report;
2) **Clarify** the results from the 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 present a web-based implementation of Visualizationary that leverages a server-side LLM, demonstrating how a "vanilla" LLM can be used for this highly complex visualization design task. Visualizationary operates as follows: We first translate chart images into text using a vision-language model (VLM) [36] and then generate automatic design guidance using an LLM as a template and text processor. Users can also upload their existing designs as an image file, leaving them free to use a tool of their own choice. The system presents the guidance in an interactive report organized hierarchically, enabling designer to drill into the findings at their desired level of detail.

In evaluation, we 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 visualizations and improve their designs. We report on four successive revisions of their visualization as well as their quantitative and qualitative feedback from 13 visualization designers of varying expertise, from a novice with only 1 year of experience in the field to an expert with 11 years of experience. The designs created

by these designers are then assessed by 3 senior visualization researchers. Our findings show the effectiveness as well as shortcomings of our concept from various perspectives, including level of expertise, type of feedback, and design optimization. These insights contribute to a deeper understanding of what constitutes effective design feedback for visualizations.

2 Background

Here we provide a brief overview of three key areas, required for understanding our work: design feedback, automating visualization design feedback, and large-language models (LLMs).

2.1 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 [34], [47], [50], [74], [75]. Munzner's nested model for visual design and validation [47] primarily incorporates this type of feedback. Conversely, peer feedback holds greater prevalence in practice, particularly for designs aimed at widespread use [1], [49], [59]. Nevertheless, this methodology has also permeated academic circles, notably within interaction design [4], [5], [14].

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 [48] 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 [12], 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.

2.2 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 [48]. This has led to a significant focus on design guidelines and rules of thumb [25], [65], textbooks with a design focus [48], and numerous resources derived from real-world practice [24]. 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 [29], visualization mirages [43], and deceptively-designed line charts [23]. For example, VizLinter [13] 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 [40] was one of the first endeavors in this area, and it was later realized in production as Tableau's Show Me feature [41]. More recent visualization recommendation methods have been proposed that generate charts using data properties [32], [72], [73], perceptual principles [72], expert feedback [39], large-scale dataset-visualization pairs [30], and design knowledge [45].

Accessibility is seeing emerging interest in the visualization field [21], [42]. Heuristic frameworks on general accessibility such as Chartability [18] could conceivably be operationalized to support automatic accessibility auditing. For color vision deficiency, Angerbauer et al. [2] present results from a large-scale crowdsourced assessment that could be similarly automated.

Finally, quantitative feedback can also help a designer improve their visualization. VisLab [16] 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 [61] 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.

2.3 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 [8], [26], [70] 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). LLMs have billions of model parameters, and thus, training these models is computationally demanding.

A standard paradigm for addressing the training costs is to pre-train LLMs on a large text corpus and then allow users to fine-tune these models on a set of desired tasks, e.g., question answering, language translation, and text summarization. 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. 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?" [36], [67]

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 of a user's visualization. These descriptions, along with user queries, are then fed into uni-modal LLMs, which generate design feedback based on the provided input. To minimize hallucination, we provide strict guidance as a preamble so that the language model does not provide random and inconsistent feedback [44].

3 Automated Visualization Feedback

We propose an automated visualization design feedback mechanism for supporting iterative design of a visualization artifact. The

designer uploads consecutive versions of a visualization artifact as they iterate on the design. The automated system then provides natural language feedback to facilitate improvement.

3.1 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. Because design critique is critical during any design process, we even find that automated design feedback is useful even for **experts** (see §6.2).

3.2 Design Rationale

Based on Munzner's visualization design and nested evaluation frameworks [47], [48], interaction design critique [4], and guidelines for visualization design [25], [45], [52], we identify several benefits for automating visualization design feedback. We explain our rationale by comparing our system's conceptual approach with the type of feedback a human designer would provide:

- R1 **Ubiquitous:** An automated system can provide immediate feedback, enabling rapid design iterations anytime [20], [22] the designer needs support. By contrast, human feedback may not always be readily-available. Furthermore, the system should not depend on the authoring tool in use.
- R2 **Real-time:** Hence, an automated visualization feedback system can provide data in a real-time, enabling designers to make immediate improvements during the design process.
- R3 **Consistency:** Automated feedback systems apply consistent evaluation criteria, reducing subjectivity and bias as well as human error in assessments. Human feedback may be less consistent and sometimes even conflicting.
- R4 **Scalability:** Compared to a human evaluator, an automated design feedback mechanism can handle a large volume of evaluations for multiple users simultaneously.
- R5 **Inexpensive:** Automating feedback can reduce (but not eliminate) human evaluators, which can be costly.

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

We propose an automated visualization design workflow called *analyze-clarify-guide-track* (ACGT) with four components:

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 § 4.4). The outcome of this analysis is a report on the visualization artifact.
2) **Clarify**: Interpreting visualization metrics on their own can be challenging [61], let alone to use as a basis for revisions. The *clarify* phase translates results from the perceptual filters into understandable terms (R3), 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 which and why certain changes are needed.
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 identified concerns. This guidance may include recommendations, actionable insights, or best practices.
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.

4 How Visualizationary Works

Visualizationary is an automated design feedback system that provides design critiques for visualization designers in natural language. The system is implemented as a web-based platform, where the front-end offers an interactive interface for users to explore various visual feedback, while the back-end supports these processes using off-the-shelf language models. Here we describe its architecture and core components in detail.

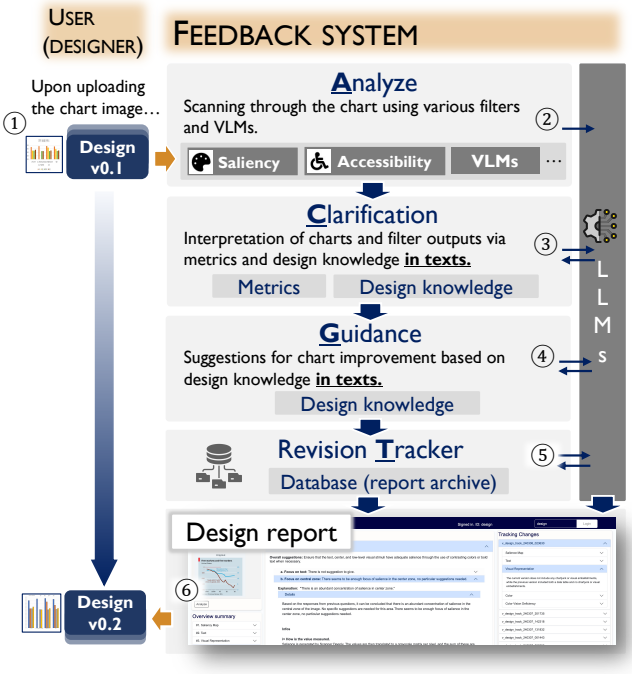

Fig. 1. **System overview.** Visualizationary is composed of a visualization design management interface and a feedback system; the white circles refer to the analysis and report steps, respectively. The system first receives a data visualization (chart) from the project container interface ① and generates 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.

4.1 System Overview

Visualizationary is a web-based client-server system (R4) that tracks a visualization artifact throughout its design process. Our system can support multiple users at once, so we placed no time constraints on image uploads (R4). However, no instances

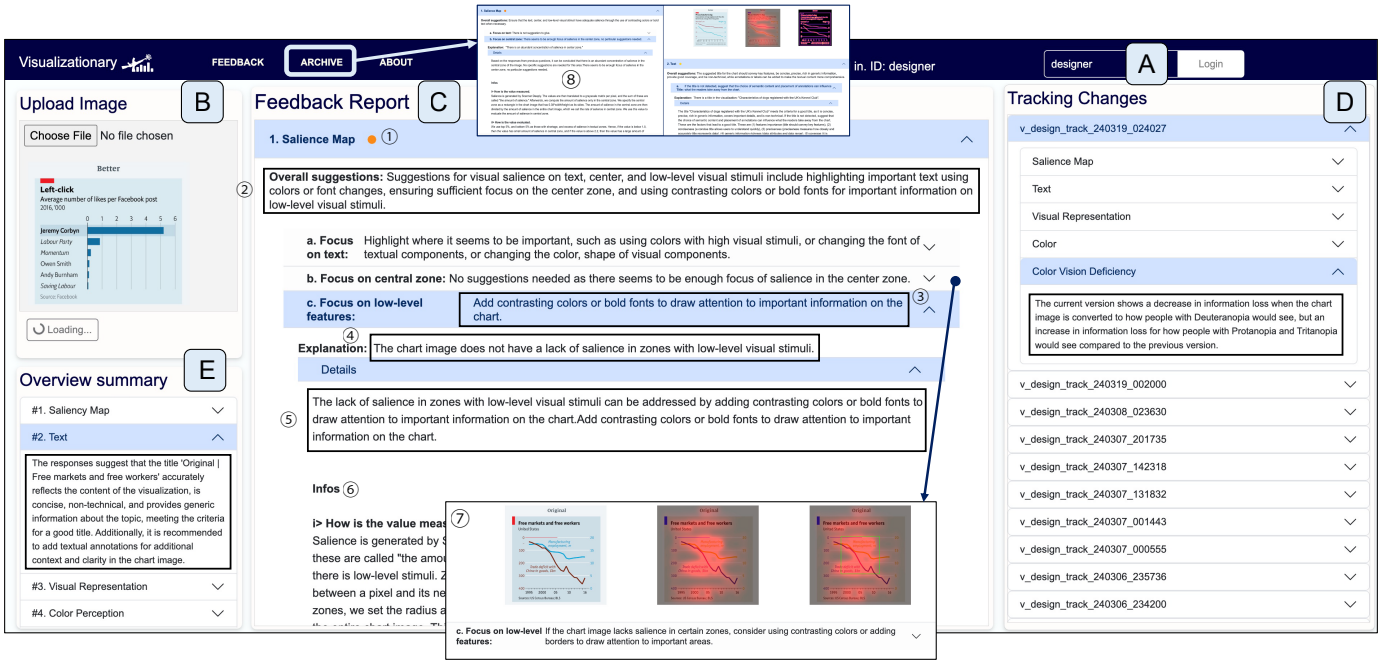

Fig. 2. **Visualizationary interface.** Example of an interactive report on a visualization artifact generated using our novel analyze-clarify-guide-track (ACGT) framework. The image is a depiction of our system, Visualizationary where a new chart image (Figure 1(B)) is being updated while the current report is about the past visualization image. The report is generated by automatically analyzing ("analyze") the artifact using an expandable 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.

occurred where multiple users accessed it simultaneously. The workflow (Fig. 1) 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 systems such as Perceptual Pat [61], which merely report metrics, Visualizationary implements the ACGT workflow from §3: beyond analyzing the image, it also explains ("clarify") the feedback using natural language (R3), suggests ways for addressing shortcomings ("guide"), and tracks the changes to the design over multiple iterations.

This workflow is realized through three components: (1) a web-based visualization design interface (R1, R4, R5) (§ 4.2), (2) interactive design reports providing feedback at user-controlled levels of detail (§ 4.3), and (3) an automated feedback system for extracting visualization data and transforming into textual feedback (R5) (§ 4.4). The former two belong to Visualizationary's frontend, whereas the third is a part of the backend. Afterwards, we detail our prompting methodologies for providing visualization feedback (§ 4.5), and our implementation notes (§ 4.6).

4.2 Visualizationary Design Management Interface

The frontend for Visualizationary is a web-based single-page application for managing the iterative design process of a visualization artifact. A *project* is a container for a design process, including all of the visualization revisions and their corresponding *design reports* (see § 4.3). The frontend has operations for creating, editing, and deleting projects associated with a user.

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. 2) (R1) and then launching the analyze-clarify-guide-track workflow on the new revision. This process typically takes a couple of minutes to complete (R2), after which a new design report is added to the timeline for the designer's perusal. In this way, the Visualizationary interface is designed to help the design process of a visualization artifact.

4.3 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 filters in § 4.4. 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. 2 (C)). Each section provides a short overall summary of suggestions (Fig. 2 (C) ②) of the underlying data for that metric followed by two standardized paragraphs:

- **Explanations:** Explanation of visualization feedback in natural language ("clarify" in the ACGT workflow) (e.g., Fig. 2 (C) ④).
- **Suggestions:** Recommendations addressing shortcomings in the design for that specific metric ("guide") (e.g., Fig. 2 (C) ③).

Expanding the subheadings yields additional explanations (see Fig. 2 (C) ⑤) and suggestions. Sections can be expanded down to raw visual metrics (see Fig. 2 (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.

Visualizationary provides additional functionality. First, 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. 2 (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. Second, to help designers quickly skim section summaries, we provide a short overview summary of responses per category in Fig. 2 (E).

We also aim to maintain transparency in the feedback process, given that both filters and LLMs can yield misleading feedback. By being transparent, designers can better determine whether to trust the model. For instance, the report shows how the system extracts chart data and calculates its metrics.

4.4 Feedback System

Here we present our feedback system (see Fig. 1). The feedback is provided using the ACGT Workflow, as explained in § 3.3. We use automated models to extract perceptual characteristics from visualizations, with the goal of providing pragmatic and actionable design feedback. These functions are a suite of computer vision and vision-language models (VLMs) that we collectively call *visualization filters*. We present a total of 10 filters from 5 topics. We first detail how these visualization filters are translated to design feedback (what corresponds to ACG of our ACGT workflow). Afterwards, we describe how the revision Tracker works.

Visualization Filters. We explain the filters according to the ACGT framework. To each filter, we provide the intended role, Analysis process, Clarification criteria, and Guidance to users. In a nutshell, the filters automatically analyze the visualizations. Then, using the information from these filters, clarification 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. Finally, based on the clarification, the LLM provides actionable guidance to users in the form of textual feedback. We instill the clarifications along with the guidance relevant to the decisions into ChatGPT as a prompt (see § 4.5). Then, based on a series of template questions, ChatGPT provides the appropriate answer.

However, despite our intents, not all filters we provide are translated into actionable feedback. This heuristic as well as examples of the feedback are described in our supplementary material. Note that the list is not exhaustive and can be extended.

Virtual Eyetracker. Virtual eyetracker refers to the extent to which areas within a scene capture the attention of an observer [55]. We utilize a virtual eyetracker [60] that predicts gaze on an overview task of a visualization image, and provide quantitative guidance. From the eyetracker, users can find where the audience would look at a visualization image as an overview. We have two reasons for providing metrics as guidance. First, we notify designers if heavy focus of salience at an area is indeed what they intended. As a case in point, if salience is primarily directed toward text, the metrics can prompt designers to reflect on whether they should shift more attention to graphical elements. Second, we alert users to potential biases by providing metrics that indicate when salience is heavily concentrated in a particular area. Because the eyetracker's performance is not perfect, it is important to inform users about possible biases, particularly when salience rates are unusually high or low. For example, the center of an image may show unusually high salience even if the area contains no graphical elements.

- *A virtual eyetracker.* **Role:** Predicts visual salience in a visualization to estimate what readers will notice during an overview task. **Analysis process:** A Scanner Deeply [60] generates an expected salience map about the inputted chart image on an overview task. **Clarification:** The salience map is overlaid as a heatmap on the chart image. The most salient areas appear in red or yellow, while less salient areas are shown in darker tones. **Guidance to users:** The salience map indicates whether the highlighted areas match the designer's intended focus [10]. It also suggests methods to improve salience in specific zones. In order to help designers focus on their areas of interests, we introduce the concept of minimizing the data-ink ratio [65], methods to highlight zones of interest (e.g., using colors, making bold, etc.), and impact of grids in charts [69].
- *Focus on text.* **Role:** Warns designers when the salience may focus overly on text. **Analysis process:** OCR software is used to detect textual zones, and texts are boxed by each letter. Then, we calculate how much salience is focused on these boxes. **Clarification:** Mark as *overly salient to text* if the salience in text boxes is above the 10^{th} percentile of the chart saliency in the dataset by Shin et al. [60]. **Guidance to users:** We warn designers that the salience of text is higher than other visualizations. We caution that the model's salience measurements may be biased or inaccurate because it tends to highlight text too readily [60]. The designer has to decide whether to believe the results or not.
- *Focus on center.* **Role:** Warns designers when the salience focuses overly on the center of the chart image. **Analysis process:** We measure the concentration of salience in the center. The center is defined as a rectangle in the middle of the image, with its height and width as $1/3\times$ of the chart. **Clarification:** A visualization is flagged as *overly salient in the center* if saliency in this region is above 10^{th} percentile. **Guidance to users:** We warn designers that the salience on the central zone [9] is higher than other visualizations. We also inform the possibility that the model's salience measurements may be biased or inaccurate.
- *Focus on visual attention.* **Role:** Warns designers when a visualization may not effectively direct visual attention to important regions. **Analysis process:** Identify areas likely to attract attention by locating zones near RGB color transitions (within a 5-pixel radius), and measure how much of these zones are covered by high saliency values. **Clarification:** A visualization is flagged as *"scarcely salient"* if saliency in these regions falls below the 90^{th} percentile of visualizations in the dataset. **Guidance to users:** Designers are alerted when attention-relevant zones receive less saliency than is typical. This matters because strong visualizations generally guide attention toward meaningful visual structures. Low saliency in such areas may reduce effectiveness.

Textual characters. Textual characters are visual objects that are used for labels, titles, and legends in visualization. Using text alongside charts can improve comprehension of visualizations [62]. To support this process, the system aims to detect title, and textual information in the chart.

- *Title.* **Role:** Detects a title and provides recommendations. **Analysis process:** Title is detected via DePlot [35]. DePlot is a VLM that translates visualizations into tables and textual components. Gathering all the information, ChatGPT pro-

vides title suggestions for the chart. **Clarification:** DePlot reads the chart image, and decides if there is a title in the image. **Guidance to users:** If a title is not detected, then recommend adding a one and provide suggestions.

– *Textual content.* **Role:** Detects text in visualization image. **Analysis process:** The OCR filter scans the chart image, and detects textual information. We use the PyTesseract-OCR as our OCR filter [64]. PyTesseract-OCR is a Python library that detects textual information from the visualization image. **Clarification:** If the OCR filter detects at least one letter in the text, then it contains the text. **Guidance to users:** When text is detected the system explains the benefits of texts in charts. If no text is detected, it explains in detail the benefits of having textual explanations in visualizations [62].

Visual representations. There exist various functionalities by knowing the chart's data and its visual representations, such as visualization linting [13], [29]. Our focus is on providing chart recommendations based on the data table and design preamble, as well as detecting unnecessary visual embellishments (chartjunk). By leveraging these suggestions, we aim to improve the visual representation of the data.

– *Optimal chart type:* **Role:** Suggests the designer an optimal chart, based on the data table translated from the chart. **Analysis process:** The data table is extracted from DePlot. The suggestion is generated based on a preamble of design knowledge and ChatGPT, which is described below. **Clarification:** The suggestions are determined based on two types of information: (1) the guideline as a preamble of design knowledge (perceptual rankings of visual cues, described in Mackinlay's APT [40]) and (2) the data table provided by DePlot. Using these two information, ChatGPT makes the final suggestion. We recognize that this is a relatively simple model with limitations, such as the inability to recommend treemaps and a restricted knowledge base for certain visualization choices. No suggestion is provided if data table, or data is not extractable from the chart image. **Guidance to users:** A chart is suggested, and ChatGPT provides both the data table and the reasoning behind its choice. This helps the designer decide whether to follow the advice.

– *Visual embellishment ("chart junk"):* **Role:** Detects any chartjunk [6] in the visualization, and if present, warns about its potential benefits and drawbacks. **Analysis process**: A computer vision model, such as YoloR [66] detects objects. **Clarification:** If the object detection model detects objects within the chart image, then we say that the model has *chartjunk*. **Guidance to users:** If chartjunk is detected, then warn of its benefits as well as dangers [6], [7].

Color perception. Color perception is a basic building block of data visualization [48], [54]. Visualization practice stipulates using a limited number of distinguishable and easily named colors [28], potentially as a function of the mark [63]. We detect the number of different and similar colors in a chart image. We then remind users about appropriate color usage. For instance, similar hues should represent continuous values, while distinct colors should signify categories [25], [48].

– *Variability of colors:* **Role:** Detects different colors and reminds the designer to check if different colors are used effectively (representing categorical values). **Analysis process:** A custom function processes the chart image to identify distinct colors. Colors are grouped as *similar* if their maximum Euclidean distance in RGB space is less than 10, with each group represented by its centroid. Each group is treated as a single distinct color. **Clarification:** If the function identifies more than 2 distinct colors from the chart image, then the image is identified as having multiple colors. **Guidance to users:** After showing all distinct colors used, the filter recommends that different colors represent categories. However, it is not capable of detecting mismatch of colors.

– *Similarity of colors:* **Role:** Detects similar colors and suggests the designer to identify if they are used effectively (representing continuous values). **Analysis process:** A customized function created in Python detects groups of similar colors (the definition is identical to similar colors described in variability of colors). A group of similar colors are determined using differences in RGB values between detected colors. **Clarification:** If the function identifies more than 2 similar groups of colors then the image is identified as having similar colors. **Guidance to users:** The filter recommends that similar colors represent continuous, related values. This filter too, is not capable of detecting color mismatch.

Accessibility. Accessibility is emerging as an important new topic in the visualization field [21], including issues such as color vision deficiency, blindness, and low vision. [15], [17], [19], [21], [42]. Although accessibility encompasses many considerations, we specifically addresses color vision deficiency (CVD). We focus on CVD because it affects a significant portion of the population (about 1 in 12 males and 1 in 200 females are affected by CVD) and directly impacts the effectiveness of color-encoded data [57].

– *CVD:* **Role:** Helps detect visualizations that are significantly affected for people with color vision deficiency (deuteranopia, protanopia, tritanopia). **Analysis process:** We evaluate the chart image from two perspectives. First, we quantitatively measure the information loss in the simulated visualizations using the entropy difference from the original image. Second, we simulate the chart image as if it were viewed with color vision deficiency. **Clarification:** If the entropy loss in the simulated visualization image exceeds a certain threshold, then the chart image is marked as *significantly affected.* **Guidance to users:** We caution that individuals with color vision deficiency may miss information. To observe the exact differences, one should view the simulated chart image. We then provide recommended CVD-safe palettes. However, it is up to the designer to select the palette.

When the system detects no quantitative issue in *clarification*, it indicates that no problem was found and then presents the filter's intent. For example, when a title is detected, the system says, "A title is detected. The intention of this filter is to determine whether a title exists and, if it does not, recommend adding one, because a title helps users better understand the visualization." Another example is when the system detects no chartjunk. In that case, the system replies as, "*We did not detect any chartjunk.*"

Revision Tracker. The *revision tracker* in Fig. 2 (D) 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 the next section. The revision tracker primarily monitors changes in quantifiable metrics for each topic. It displays any increases or decreases and provides commentary on whether those changes are beneficial. Because it focuses on metrics, it does not capture qualitative comparisons (e.g., saliency maps or simulated CVD images). To address that gap, we also offer a detailed tracking capability. For detailed tracking, Visualizationary allows designers to check all versions of their past reports by going to 'Archives' in the navigation bar (see Fig. 2 ⑧). The interface provides two separate screens, each displaying a report of past version selected by the user, to facilitate the comparison process.

4.5 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* [3], [53], [70], [71], [76]. Prompt engineering is typically based on empirical data specific 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 [37] in designing prompts to avoid such non-deterministic and long-winded behavior:

- *Use exact keywords:* use concepts as reserved words;
- *No synonyms:* varying labels yields nondeterminism;
- *Avoid unnecessary data:* irrelevant data expands scope;
- *Constrain feedback length:* avoid unfocused answers;
- *Ask specific queries:* limit the scope of the query; and
- *Avoid open-endedness:* encourage short/direct responses.

To further minimize hallucinations, we incorporate relevant grounding contexts (e.g., design knowledge about data visualizations and filters) [44] in the prompt, preventing the LLM from generating text outside its learned scope.

We construct two types of prompts: *analyze-clarify-guide* prompts and *track* prompts. To create those prompts, we define the following templates T (full prompts are included in OSF [1]):

ACG-template. We call the analyze-clarify-guide (ACG) prompt template T_{acg}. `[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 `[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 `[filter-suggestions]`, a compact representation of visualization design knowledge in natural language; see the supplementary material.

We mainly use "if, then" tone when providing design knowledge. While this may not be the most natural style of discussion, we want to convey that our suggestions are optional and to prevent designers from applying feedback inaccurately.

[1]https://osf.io/v7hu8

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.

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

4.6 Implementation Notes

The Visualizationary system is implemented as a web application using HTML, CSS, JavaScript, and JQuery. We use the Python Flask web framework[2]; the analysis components were implemented server-side in Python 3. We store data into MongoDB.[3] During the user study, Visualizationary was hosted on Amazon Web Services (AWS). We use `gpt-3.5-turbo` as LLM.

5 User Study

We performed a user study to assess the impact of LLM-generated feedback in the visualization design pipeline using the Visualizationary system. We opted for a qualitative study where designers of different skill were asked to use Visualizationary to support the design process of a new visualization over several days. We then let a small group of visualization experts 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).

5.1 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 a year. As a result, we were able to recruit 13 visualization designers of different seniority. Table 1 shows the demographics of the participants. These participants were paid a $40 Amazon gift card upon concluding the study. While Visualizationary was originally designed for novice and intermediate visualization designers, our

[2]https://flask.palletsprojects.com/
[3]https://www.mongodb.com/

TABLE 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 a year of experience in designing visualizations. We categorize those with less than or equal to 3 years of experience in visualization as novice (6 participants, N1-N6), those with 4 to 7 years as intermediate (4 participants, I1-I4), and those with more than 7 years as expert groups (3 participants, E1-E3), respectively. We present participants' gender, age, education (Edu), job title, and years of experience in Vis (Exp.).

ID	Gender	Age	Edu.	Job Title	Exp.
N1	Male	33	B.S.	Software engineer	1 yr
N2	Female	33	B.S.	Primary school teacher	1 yr
N3	Male	32	Ph.D.	Postdoc in CS	1 yr
N4	Male	26	B.S.	Ph.D. student in CS	3 yrs
N5	Male	29	M.S.	Software engineer	3 yrs
N6	Female	24	B.S.	M.S. student in Data Science	3 yrs
I1	Male	30	B.S.	Ph.D. student in CS	5 yrs
I2	Male	34	Ph.D.	Research scientist	5 yrs
I3	Male	28	M.S.	Ph.D. student in CS	6 yrs
I4	Male	31	M.S.	Ph.D. student in CS	6 yrs
E1	Male	31	M.S.	Data scientist	9 yrs
E2	Male	33	Ph.D.	Assistant professor in Vis	10 yrs
E3	Male	35	Ph.D.	Postdoc in Vis	11 yrs

participants also included expert ones. We included such participants because we were interested in seeing whether our system could benefit even an expert.

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.

5.2 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:

- They must spend at least two hours on the design process;
- They should work independently on the design task;
- Image size between 100×100 and 2000×2000 px;
- Images 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 publication.

5.3 Procedure

The study was performed exclusively online using video conferencing and consisted of four steps. We obtained participant demographic information and scheduled the dates for each step during recruitment. Here we describe 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 (§ 5.2). 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 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 (R2) 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 (R1). During this period, the experiment administrator was available to answer technical questions.

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's each iteration (that is, 5 ratings and 5 assessment sentences in total).

5.4 Data Collection

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

5.5 Apparatus

The interview and the experiment were conducted using the participants' own computers. Although we did not enforce strict hardware constraints, we recommended participants use a resolution of at least 1920×1080 and 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.

6 Results

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

TABLE 2
Participants' study setup. Here we list the settings the participants used to conduct our study. We present the tools, datasets and chart types used by the participants. We also show the total time (hh:mm) taken for these participants to finish the work (all participants updated five designs within a single continuous time frame).

ID	Tool Used	Dataset Topics	Chart Type Explored	Time Between V1 and V5
N1	Microsoft Excel	Time management	Stacked area/bar charts, line chart	01:47
N2	Microsoft Excel	Survey analysis	Bar chart	01:23
N3	Python (Matplotlib)	Topic document analysis	Bar chart	01:14
N4	Python (Matplotlib)	Scientific result analysis	Bar chart, line chart	02:16
N5	Python (Matplotlib)	Dataset analysis	Line chart, bar chart	01:48
N6	Tableau	Scientific result analysis	Dot plot, line chart	02:31
I1	Javascript (D3.js)	Cluster analysis	Matrix scatterplot	01:54
I2	Tableau	Credit card usage	Bar chart, Stacked bar chart	02:28
I3	Microsoft Excel	User experience on UI	Bar chart	03:18
I4	Microsoft Excel	Scientific result analysis	Bar chart, stacked bar chart	02:04
E1	Tableau	Sales data analysis	Geographic map, bubble chart	02:35
E2	Spotfire	Market analysis	Bar chart	02:15
E3	Microsoft Excel	Scientific result analysis	Boxplot	01:47

TABLE 3
Main feedback taken by participants. Here we summarize the list of feedback mainly taken by participants. We detail the type, and detail the categories (CHART, SAL, COL, TEXT, and CVD) of, and filters (–REC (recommendation), –CENTER/TEXT/POI MISMATCH (focus on center/focus on text/mismatch of salience patterns), TITLE/TEXT (no text/no title), and SIM/VAR (similarity, variability of colors) themselves used per iteration. As a case in point, Iteration 1 refers to the feedback taken by the participant to create their second version.

ID	Iteration 1	Iteration 2	Iteration 3	Iteration 4
N1	CHART–REC	SAL–CENTER	SAL–CENTER	SAL–POI MISMATCH
N2	SAL–CENTER	CVD, SAL–CENTER	CVD, SAL–CENTER	SAL–POI MISMATCH
N3	SAL–CENTER, COL–VAR	SAL–CENTER, CVD	SAL–POI MISMATCH	SAL–TEXT
N4	COL–SIM, TEXT–TITLE	CVD	SAL–POI MISMATCH	SAL–POI MISMATCH
N5	COL–VAR	TEXT–TITLE	SAL–POI MISMATCH	CVD
N6	CHART–REC	SAL–POI MISMATCH, TEXT–TITLE	TEXT–TEXT	COL–VAR, SAL–POI MISMATCH
I1	TEXT–TITLE, SAL–CENTER	SAL–POI MISMATCH, CHART–REC	SAL–POI MISMATCH	SAL–POI MISMATCH
I2	SAL–TEXT	SAL–TEXT	SAL–TEXT	SAL–TEXT
I3	COL–SIM, TEXT–TEXT	COL–SIM	CVD	CVD, SAL–CENTER
I4	CHART–REC	TEXT–TITLE, SAL–POI MISMATCH	SAL–CENTER, COL–VAR	SAL–POI MISMATCH
E1	CVD	SAL–POI MISMATCH	TEXT–TEXT	SAL–POI MISMATCH
E2	CHART–REC	TEXT–TITLE	SAL–POI MISMATCH, TEXT–TITLE	TEXT–TEXT, SAL–POI MISMATCH
E3	TEXT–TEXT, TEXT–TITLE, SAL–CENTER	COL–VAR	COL–VAR	SAL–POI MISMATCH

6.1 Design Evolution

Here, we describe how the participants' designs evolved. First, we provide an overview of their experimental setup. Table 2 details the tools used, dataset themes, chart types, and the time spent iterating each design. We found that a variety of tools (6 in total) are employed regardless of expertise level, and the datasets covered diverse topics. This justifies our decision to upload visualizations as image files, since this approach keeps the feedback independent of the tool. Bar charts were the most common chart type, adopted by eight participants, though nine different visualization types appeared in total. The time spent is calculated as the interval between the first and last upload. While we cannot determine exactly how much effort was allocated to each design within this period, a few patterns emerged. Participants worked in one long session, and it lasted approximately 90 to 150 minutes.

Second, we present information about participants' design evolution processes. Figs. 3 and 4 show visualization snapshots from all 13 participants, categorized into novice, intermediate, and expert groups. Larger snapshots and full descriptions appear in the supplementary material. During interviews, we asked participants to explain the changes they made in each iterative step and to discuss their intentions. Below we discuss the design evolution for three selected participants, N1, I2, and E1 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. Results for participants not covered here can be found in the supplementary material.

Participant N1 (novice). N1 created a stacked area chart (on Microsoft Excel's recommendation) to visualize daily activities.

- **Version 1→2:** *Feedback:* Visualizationary recommended bar charts to enhance value comparisons. *Revision:* N1 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. *Revision:* N1 adjusted the chart's size ratio and incorporated a new line chart, which displayed the cumulative average hours spent on a specific activity ("work"). N1 introduced a vertical axis to represent the line value for examining influences on work.
- **Version 3→4:** *Feedback:* Again, Visualizationary observed excessive concentration of salience in the central area. *Revision:* N1 re-adjusted chart dimensions and highlighted labels.
- **Version 4→5:** *Feedback:* Saliency is focused in unimportant areas. *Revision:* N1 increased title and legend font sizes.

Participant I2 (intermediate). I2 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:* Visualizationary asserted that there

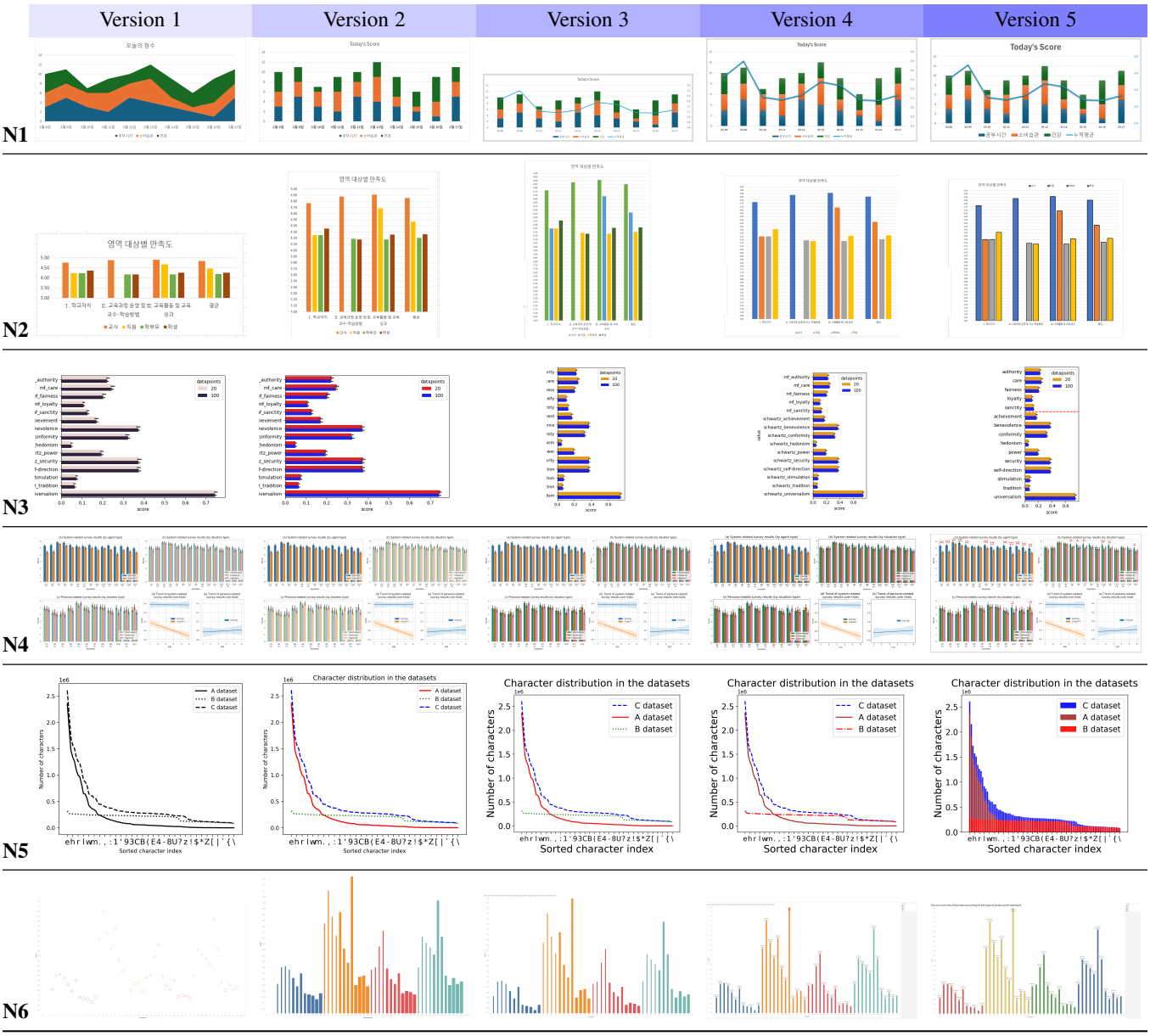

Fig. 3. **Examples of visualization design evolution (Novices).** These images represent the five most significant versions (as identified by the participants) of novice visualization designers during the design process of creating a new visualization artifact from scratch. The explanations of these design iterations can be found in the supplemental material.

is too much focus on text. *Revision:* I2 changed the order of the hierarchy to remove repetitive labels.

- **Version 2→3:** *Feedback:* Again, Visualizationary stated that there is too much focus on text. *Revision:* I2 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:* I2 eliminated the text in the bar chart and introduced a legend.
- **Version 4→5:** *Feedback:* And again, Visualizationary maintained that there is still too much text. *Revision:* I2 rotated the bars vertically and separated the category labels.

Participant E1 (expert). Using Tableau, E1 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.
- **Version 2→3:** *Feedback:* The salience map in Visualizationary directed attention to country names, but several were unrelated to the data. *Revision:* Country names were added to each circle, and irrelevant country names were removed.
- **Version 3→4:** *Feedback:* Visualizationary noted that textual content would make the chart more readable. *Revision:* E1 relocated country names to circle centers, added explanatory boxes, and differentiated ocean colors for clarity.
- **Version 4→5:** *Feedback:* Scanner Deeply did not detect some circles. *Revision:* Strokes were added to circles.

6.2 Feedback Usage Patterns

Below we present a summary of feedback usage patterns. Table 3 outlines the filters each participant used during the design process. To begin with, we describe iteration patterns, and then we also analyze cases when participants do not take the feedback.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2025.3579700

Fig. 4. **Examples of visualization design evolution (Intermediate and expert designers).** These images represent the five most significant versions (as identified by the participants) of intermediate and expert visualization designers during the design process of creating a new visualization artifact from scratch. The explanations of these design iterations can be found in the supplemental material.

Iteration patterns. Across the 13 participants, each design iteration typically incorporated one or two feedback changes, occasionally reaching three in a single iteration. Of the 68 total feedback instances, 33 were related to "Scanner Deeply," with 18 of these examining whether salience aligned with the designer's intended focal points. Notably, salience-related feedback often required multiple iterations to complete, possibly because shifting salience in a single step is not straightforward. Chart recommendations appeared only five times—primarily in the first iteration. This seems to reflect the tendency to make major design choices (e.g., chart type) prior to making minor design choices [58]. The chartjunk filter was never triggered, as no chartjunk elements were introduced in the participants' designs. Although usage patterns were relatively consistent across skill levels, a slight decrease in implemented feedback was observed from early to late iterations (an average of 1.46 changes in the first iteration versus 1.23 in the last), which may indicate that designs converged over time, reducing the need for additional modifications.

When participants do not listen to the feedback. We observe several instances where participants noted the system's feedback but ultimately chose not to implement it. The most frequent reason was that the model did not fully align with the user's intent. When I4 was advised to use additional colors for separate categories, they declined because only two colors were necessary for their comparative task. In other cases, users felt the system's suggestions conflicted with their design objectives. For instance, N4 wanted to include a grid for comparison purposes, despite realizing it might reduce salience in key areas. Some participants also proceeded

TABLE 4

Expert evaluation of visualization designs. We deployed 3 seasoned visualization experts to evaluate visualization designs authored by 13 participants. The scores from second to fifth columns show the evaluation of visualization designer per iteration. We present the trends of scores per participant in the sixth column. We also asked the experts to pick the best design from each designer's iterations. This is shown in the seventh column. Finally, the last column shows the overall improvement scores of visualization designs. The three numbers in brackets in the third and fifth column represent the scores obtained by the experts, sorted per different expert evaluator. The 1–5 Likert scale is defined as follows: 1 = significant decline in quality, 3 = neutral, and 5 = significant improvement in quality.

ID	Expert evaluation scores per iteration				Trends	Votes for the best version	Overall score (V1 - V5)
	V1 → V2	V2 → V3	V3 → V4	V4 → V5			
N1	3.00 (3/4/2)	3.00 (5/2/4)	3.67 (3/4/4)	3.33 (3/3/4)	▬ ▬ ▬ ▬	1 1 1	3.67 (4/3/4)
N2	2.33 (4/1/2)	2.67 (3/2/3)	3.00 (4/3/2)	3.00 (4/3/2)	▬ ▬ ▬ ▬	2 1	2.67 (3/3/2)
N3	3.33 (3/3/4)	1.67 (1/2/2)	4.67 (4/5/5)	3.33 (4/2/4)	▬ _ ▬ ▬	1 2	4.00 (4/4/4)
N4	3.00 (3/4/2)	2.67 (2/3/3)	2.67 (3/3/2)	3.00 (4/3/2)	▬ ▬ ▬ ▬	1 1 1	2.67 (3/3/2)
N5	4.33 (4/4/5)	2.33 (3/2/2)	3.00 (4/1/4)	3.00 (2/3/4)	▬ _ ▬ ▬	1 2	3.00 (3/3/3)
N6	4.33 (4/4/5)	4.33 (4/5/4)	4.00 (3/5/4)	4.00 (4/4/4)	▬ ▬ ▬ ▬	3	4.00 (4/4/4)
I1	3.67 (3/4/4)	3.00 (4/3/2)	2.67 (2/3/3)	3.33 (3/4/3)	▬ ▬ ▬ ▬	1 1 1	2.67 (3/3/2)
I2	4.33 (4/4/5)	4.00 (5/3/4)	3.67 (2/4/5)	2.33 (2/3/2)	▬ ▬ ▬ _	3	3.67 (4/4/3)
I3	3.33 (4/2/4)	3.33 (4/3/3)	2.33 (1/3/3)	5.00 (5/5/5)	▬ ▬ _ ▬	3	5.00 (5/5/5)
I4	4.00 (4/5/3)	4.00 (4/4/4)	4.33 (5/4/4)	4.33 (3/3/4)	▬ ▬ ▬ ▬	2 1	4.33 (5/4/4)
E1	3.00 (2/3/4)	4.33 (4/5/4)	4.00 (5/3/4)	3.67 (3/4/4)	▬ ▬ ▬ ▬	3	4.67 (4/5/5)
E2	3.00 (2/5/2)	3.33 (3/3/4)	4.33 (5/4/4)	4.33 (4/4/5)	▬ ▬ ▬ ▬	3	3.00 (1/4/4)
E3	4.33 (4/4/5)	3.67 (3/4/4)	4.67 (5/5/4)	3.00 (3/3/3)	▬ ▬ ▬ ▬	2 1	4.67 (4/5/5)

TABLE 5

Feedback evaluation per filter types. We present the average feedback score for each filter type, based on iteration-level scoring. We also provide a summary of the effectiveness and limitations of the filters used in Visualizationary per topic, based on the comments provided by the expert evaluators. The numbers in brackets indicate the corresponding count of feedback instances for each score. Overall, designers improved when using the feedback, although novices found the CVD-related feedback more challenging to apply.

	Novices	Intermediates	Experts	Effectiveness	Limitations
Virtual eyetracker SAL	3.06 (12)	3.79 (11)	4.00 (7)	Helps designers highlight key parts	Task-specific saliency prediction
Text TEXT	3.28 (6)	4.00 (4)	4.08 (4)	Helps add titles and explanations	Does not suggest text contents
Visual representation CHART	3.44 (3)	4.00 (2)	3.00 (1)	Helps recommend simple charts	Does not recommend complex charts
Color perception COL	3.67 (4)	3.66 (3)	4.17 (2)	Guides proper color use by data type	Color choice relies on the designer
CVD CVD	2.60 (5)	3.67 (2)	3.00 (1)	Helps be aware of CVD	Color choice relies on the designer

with their own experiments unrelated to any feedback, such as N1 adding a line chart to meet personal objectives. Finally, another example is when the system presents incorrect recommendations. For instance, participant I1 was advised to replace a heatmap with a bar chart but did not know how to create this chart for their data.

Overall, these observations indicate that the system is not as effective when users have their own objectives, as it currently lacks the capacity to capture their intentions. They also reveal that some filters within the system are not optimal.

6.3 Design Evaluation

We here present the results of the design assessment by external reviewers, which is the final step of our study procedure (see § 5.3). We recruited three senior visualization researchers who all have more than 7 years of experience to evaluate the designs conducted by the participants. Table 4 shows various evaluations on the improvement of visualization designs conducted by our participants. Columns from second to fifth show the evaluation of visualization designer per iteration. The sixth column shows the increasing/decreasing trend of the scores per participant. The seventh column shows which design the experts judged to be the best among each designer's iterations. Finally, the last column shows the overall improvement scores of visualization designs. Note that the values in the second to fifth and the last columns are not on the quality of the design but on the improvement of the design. Also, we analyze the evaluation scores, effectiveness, and limitations per feedback types. This is shown in Table 5.

On the whole, evaluators largely believed that the participants' designs improved from their starting point, averaging a score of 3.69 (3 is neutral). N2, N4, and I1's designs were modestly below the neutral mark. Evaluators observed consistency in N4's designs, with a slight decrease in color quality. For I1, while adding a title was beneficial, the chart's readability declined. N2 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. Below, we provide detailed analyses of the tables.

When feedback is effective. We identified several ways in which the system's feedback proved effective. First, as designers gained more expertise, they used the salience feature more effectively, which led to an improved data ink ratio. This was also noted by the evaluators. Most issues raised by the system were resolved within one or two iterations, although salience maps sometimes required additional effort. Even novice users managed salience related challenges by comparing designs across iterations and using the tracker interface (Fig. 2 (D)) together with archives for more detailed heatmap comparisons. In addition, feedback on text placement and color usage led to tangible improvements.

These findings indicate that our ACG feedback framework, as well as feedback in textual form, was effective across various skill levels. We describe the role of the tracker interface (T) in § 6.4.

When feedback is not effective. Although the system demonstrated effectiveness in many situations, it was not universally successful. One reason is that novices did not receive feedback that was sufficiently direct or actionable. In practice, novices had difficulty adapting color palettes for color vision deficiencies, which indicates that selecting and applying a limited set of colors can be challenging for less experienced designers. Some also struggled with salience-related tasks, such as precisely shifting attention to specific chart elements. In the future, we plan to explore more accessible ways to recommend color palettes for CVD and to provide more direct salience-related feedback.

Furthermore, the system focuses on general communicative purposes, and thus does not yet account for specialized design intentions. For example, I1 prioritized data analysis rather than optimizing the chart for communicative goals. Although the expert evaluators were unaware of this, it may still have been appropriate for the designer's specific objectives.

Moreover, we observe cases where the filters in the system occasionally malfunction. For example, salience may appear concentrated in a region with no chart elements (N6), or, if the chart is too complex (I1), Deplot might fail to read the data correctly and recommend an inappropriate chart. Although such cases are relatively not common, they reduce confidence in the filters.

Best version from iterations. To determine which iteration yielded the most polished design, we asked evaluators to select the best version out of five. Our overview shows that when a design's score is lower, evaluator opinions are more divided, whereas a higher score reflects consensus favoring the fifth version. However, the final design is not always the best for several reasons. First, if feedback-driven changes do not result in noticeable improvements, evaluators may choose different versions. For instance, in participant I1's case, the design changes were minor and barely discernible, leading three evaluators to select three different versions. N4 exhibited a similar pattern. Second, designers' goals can sometimes introduce unintended changes that degrade the overall design from an outside perspective. For example, near the end of participant E3's process, an element became overly highlighted; evaluators, unaware of this intention, did not favor that final iteration. Third, when changes are trivial, observers may not detect a meaningful difference between one iteration and the next. For example, in participant N3's final iteration, only a single red grid line was added to enhance clarity, which did not substantially shift evaluators' perceptions.

Overall, when design changes are clear and demonstrably beneficial, evaluators' judgments tend to converge. Conversely, if the changes are less distinct or their benefits uncertain, their evaluations diverge. We also observe that some designers have their own goals, and evaluators assessed them as a communicative visualization without knowing the designers' intents.

Variations in evaluators' scores. In some cases, the evaluators' scores converged, yet there were also instances of considerable divergence. Overall, the variance in final scores was generally minimal: most evaluators agreed or differed only slightly. However, participant E2 presented an exception, where two evaluators gave a score of 4 and one assigned a 1. The evaluator who gave a 1 stated, *"I do not see how changing from a heatmap to a bar chart is an improvement. I might even prefer the heatmap."* This discrepancy appears related to the communicative aspect of visualization, in which the evaluator's limited understanding of the designer's intentions played a role.

We observed even greater variations in the evaluators' assessments across different iterations. Despite having solid expertise in visualization, each evaluator brought unique preferences and standards for communicative visualization. For example, moving from V2 to V3 in I2's design, one evaluator was neutral about introducing multiple colors (assigning a 3), another greatly approved (giving a 5), and a third specifically praised the altered chart type (awarding a 4). Similar disagreements arose when some evaluators judged the overall positive effect of a design change, whereas others compared only the previous and current versions. One evaluator noted, *" To judge whether a change is good or bad, we need to consider the entire sequence of modifications. Design is not always a greedy process. Can we dismiss a change as having low value just because it looks unfavorable in one instance?*"

These findings underscore the complexity of evaluation when preferences vary. They also highlight that evaluators, unfamiliar with the designer's intention, may reach divergent conclusions.

6.4 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. For every design refinement, we asked participants to mainly elaborate on 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. Detailed below are the summarized comments.

Role of the feedback. Visualizationary not only detects issues but also provides expert-level knowledge that many participants had not initially considered. In some cases, the system highlighted overlooked concerns, such as color deficiency filters. N4 noted, "*[the system] is capable of showing problems that I was not aware of, although I should have been,*" while E2 remarked, *"even experts may not be aware of those specific details about visualization principles."* However, the system occasionally misdiagnosed the chart image, leading to feedback that was out of context.

Moreover, Visualizationary could provide a more detailed, actionable solutions, as some complex tasks still required deep human decisions. Although all participants attempted to address the system's feedback, some novice- and intermediate-level designers found it difficult to implement suggestions that were not specific enough, such as redistributing saliency attention. For instance, I2 devoted every iteration to balancing a chart's salience, and I1 sought to focus salience on a particular area, underscoring the varying difficulty of applying automated feedback.

Role of tracker. The consistent use of the tracker interface (T) for iterative refinements, especially when a problem could not be resolved in a single step, further underscores its supportive role in the design process. We find that most participants that use virtual eyetracker have the experience of using the tracker interface to track their improvements (11/13). For people that use saliency maps, it functioned as a comparison system that measures whether there were improvements between the current and the past version. I2, 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." Furthermore, tracking changes per iteration allowed participants to catch up (3/13).

There were also a few participants who did not use tracking. They uploaded the designs one after another without much delay. N4 was an example, noting *"Because I remember what needs to be changed, I did not look into the system."*

Role of the textual report. Because the output of this report is generated by an LLM, 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 drill down to the important details. 10 out 13 participants felt that this capability helped them be more organized, though some still felt overwhelmed at the volume of text.

General assessment. Participants considered the system an assistant providing expert feedback. Many people originally thought that it would be difficult to provide useful insights, but were surprised at its capability to help the design. Furthermore, the use of an LLM created a sense of a "human" response, which made some participants feel less isolated during the design process.

However, some areas could be enhanced. One key observation was that, despite the hierarchical structure, participants still found the text volume overwhelming.

7 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) our ACGT framework is effective for providing feedback for communicative visualizations. Below we discuss issues on visualization design feedback, as well as the limitations of our work.

7.1 Visualization Design Feedback

Bridging design contexts. Although we designed our feedback primarily to enhance clarity in visualizations, it is equally important to consider each designer's unique goals and intentions. Some participants diverged from the system's advice because it did not align with their specialized needs (see § 6.3). This underscores that relying solely on general "best practices" can overlook the nuances of personal or domain-specific objectives.

Balancing clear, communicative design with each user's specialized goals remains a significant challenge. On one hand, emphasizing universal guidelines can neglect individual contexts; on the other, tailoring feedback for every nuanced purpose requires more extensive modeling of the designer's intent. We see this as an important avenue for future work, where feedback systems can be refined to address both communicative effectiveness and the broader range of a designer's motivations.

Human-like visualization feedback. AI models can help designers assess whether a visualization is "good enough" by applying best practices and accessibility guidelines. Systems like Visualizationary use automated checklists to approximate human perception, for instance by evaluating color vision deficiency and visual saliency. One way to enhance this guidance is to add more filters tailored to designers' needs. To further validate a design or create domain-specific conditions, designers can set benchmarks (e.g., speed of comprehension) to measure success.

Nevertheless, visualization design is context-dependent and subjective, as it involves aesthetic considerations and domain-specific requirements. Can we develop a subjective design evaluator that declares a design "complete" and provides feedback in a style akin to a human visualization designer? This would not only guide users to general feedback, but also provide design feedback that would be tailored to their owners' styles. Achieving this would require extensive knowledge of best practices, accessibility standards, and real user feedback. As a possible future direction, we plan to explore how effectively LLM-driven agents could contribute to building such models.

Pitfalls of automated chart authoring. One important goal for Visualizationary was to help novice designers create visualizations. Our results indeed show that our system 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, we speculate that automation and lack of fundamental design skills could be contributing factors.

By way of explanation, note that 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). N2, 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."* N1 shared similar comments. While automated tools have simplified visualization creation, their influence may have affected our results in Table 4.

7.2 Limitations

Avenues for future improvements. Our system is designed to enable a variety of future research avenues. To begin with, it can accommodate additional filters—for example, detecting charts that do not begin at zero (as in N2's case) and introducing linting-like features to ensure consistency. Second, accurately predicting a saliency map for an overview task is challenging, as saliency varies by task and does not always align with overview-oriented goals. This presents the challenge of predicting saliency based on the user's intent [68]. Third, more advanced interpretability for each filter could yield more actionable feedback for designers. For instance, refining saliency-based methods may help pinpoint issues more precisely, and automating which part of the visualization the feedback refers to would streamline problem identification. Fourth, enhancing our system's ability to deliver user-friendly and clear feedback further expands its potential by helping designers better understand the feedback. For example, by going beyond the "if, then" structure in our design guidelines (as noted in § 4.5) and exploring alternative tones, we can improve how feedback is delivered. We can also refine our experimental procedures by adopting more neutral language (e.g., "What was your overall impression of the system?") rather than "Did you like the system?" to mitigate bias. While we cannot address every challenge at once, we leave these avenues open for continued exploration.

Hallucination. Despite their capabilities, LLMs may produce nonsensical or unfaithful content, commonly referred to as *hallucinations* [31]. Hallucinations may lead to feedback that contradicts the designer's original goals or fails to improve the visualizations. This issue poses a potential risk to Visualizationary, as users build

trust in the models powering our system when most of their outputs are consistently reliable. Recent work has proposed defensive methods to mitigate hallucinations, such as incorporating a user's feedback into the model's fine-tuning [33] and detecting self-contradiction in model-generated texts [46]. Incorporating these methods remains an avenue for future work.

Privacy risks. Because our system are built on LLMs, one can be concerned about the associated privacy risks, such as membership inference or data extraction attacks [11]. User data may be intentionally (or unintentionally) leaked to an adversary with access to these models [27]. The concerns primarily stem from two aspects: (1) user data being shared with commercial chat-based services and (2) the fine-tuning of these models on user data. However, the first concern can potentially be mitigated by avoiding the use of third-party hosted LLMs. We can entirely prevent our system from accessing external services by employing local open-source models. Second, we do *not* fine-tune the models we use on user data, ensuring that user data cannot be memorized by our LLMs or leaked by malicious actors with access to the models.

8 Conclusion

We have presented Visualizationary, a system 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 system. 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.

Acknowledgments

We thank anonymous reviewers for their feedback. This work was partly supported by grant IIS-1901485 from the U.S. National Science Foundation (Shin), a Google Faculty Research Award (Hong), and Villum Investigator grant VL-54492 by Villum Fonden (Elmqvist). Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] L. Alabood, Z. Aminolroaya, D. Yim, O. Addam, and F. Maurer. A systematic literature review of the design critique method. *Information and Software Technology*, 153:107081, 2023.

[2] K. Angerbauer, N. Rodrigues, R. Cutura, S. Öney, N. Pathmanathan, C. Morariu, D. Weiskopf, and M. 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*, pp. 134:1–134:23. ACM, New York, NY, USA, 2022. doi: 10.1145/3491102.3502133

[3] S. Arora, A. Narayan, M. F. Chen, L. J. Orr, N. Guha, K. Bhatia, I. Chami, F. Sala, and C. Ré. Ask me anything: A simple strategy for prompting language models. *CoRR*, abs/2210.02441, 2022. doi: 10.48550/arXiv.2210.02441

[4] J. Bardzell. Interaction criticism: An introduction to the practice. *Interacting with Computers*, 23(6):604–621, 2011. doi: 10.1016/j.intcom.2011.07.001

[5] J. Bardzell, J. D. Bolter, and J. Löwgren. Interaction criticism: three readings of an interaction design, and what they get us. *Interactions*, 17(2):32–37, 2010. doi: 10.1145/1699775.1699783

[6] S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. 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*, pp. 2573–2582. ACM, New York, NY, USA, 2010. doi: 10.1145/1753326.1753716

[7] 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. doi: 10.1109/TVCG.2013.234

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901. Curran Associates, Inc., Red Hook, NY, USA, 2020.

[9] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand. What do different evaluation metrics tell us about saliency models? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(3):740–757, 2019. doi: 10.1109/TPAMI.2018.2815601

[10] Z. Bylinskii, N. W. Kim, P. O'Donovan, S. Alsheikh, S. Madan, H. Pfister, F. Durand, B. Russell, and A. Hertzmann. Learning visual importance for graphic designs and data visualizations. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, p. 57–69. ACM, New York, NY, USA, 2017. doi: 10.1145/3126594.3126653

[11] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *Proceedings of the USENIX Security Symposium*, pp. 267–284. USENIX Association, Berkeley, CA, USA, 2019.

[12] S. K. Chandrasegaran, S. K. Badam, L. G. Kisselburgh, K. Ramani, and N. Elmqvist. Integrating visual analytics support for grounded theory practice in qualitative text analysis. *Computer Graphics Forum*, 36(3):201–212, 2017. doi: 10.1111/CGF.13180

[13] Q. Chen, F. Sun, X. Xu, Z. Chen, J. Wang, and N. Cao. VizLinter: A linter and fixer framework for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):206–216, 2022. doi: 10.1109/TVCG.2021.3114804

[14] R. Cheng, Z. Zeng, M. Liu, and S. Dow. Critique me: Exploring how creators publicly request feedback in an online critique community. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2), 2020. doi: 10.1145/3415232

[15] J. Choi, S. Jung, D. G. Park, J. Choo, and N. Elmqvist. Visualizing for the non-visual: Enabling the visually impaired to use visualization. *Computer Graphics Forum*, 38(3):249–260, 2019. doi: 10.1111/cgf.13686

[16] J. Choi, C. Oh, Y.-S. Kim, and N. W. Kim. VisLab: Enabling visualization designers to gather empirically informed design feedback. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 813:1–813:18. ACM, New York, NY, USA, 2023. doi: 10.1145/3544548.3581132

[17] P. Chundury, B. Patnaik, Y. Reyazuddin, C. Tang, J. Lazar, and N. Elmqvist. Towards understanding sensory substitution for accessible visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):1084–1094, 2022. doi: 10.1109/TVCG.2021.3114829

[18] F. Elavsky, C. L. Bennett, and D. Moritz. How accessible is my visualization? evaluating visualization accessibility with chartability. *Computer Graphics Forum*, 41(3):57–70, 2022. doi: 10.1111/cgf.14522

[19] F. Elavsky, L. Nadolskis, and D. Moritz. Data navigator: An accessibility-centered data navigation toolkit. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):803–813, 2024. doi: 10.1109/TVCG.2023.3327393

[20] N. Elmqvist. Data analytics anywhere and everywhere. *Communications of the ACM*, 66(12):52–63, 2023. doi: 10.1145/3584858

[21] N. Elmqvist. Visualization for the blind. *Interactions*, 30(1):52–56, 2023. doi: 10.1145/3571737

[22] N. Elmqvist and P. Irani. Ubiquitous analytics: Interacting with big data anywhere, anytime. *IEEE Computer*, 46(4):86–89, 2013. doi: 10.1109/mc.2013.147

[23] A. Fan, Y. Ma, M. Mancenido, and R. Maciejewski. Annotating line charts for addressing deception. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 80:1–80:12. ACM, New York, NY, USA, 2022. doi: 10.1145/3491102.3502138

[24] S. Few. *Information Dashboard Design: The Effective Visual Communication of Data*. O'Reilly Press, Sebastopol, CA, USA, 2006.

[25] S. L. Franconeri, L. M. Padilla, P. Shah, J. M. Zacks, and J. Hullman. The science of visual data communication: What works. *Psychological Science in the Public Interest*, 22(3):110–161, 2021. doi: 10.1177/15291006211051956

[26] D. Ganguli and et al. Predictability and surprise in large generative models. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, pp. 1747–1764. ACM, New York, NY, USA, 2022. doi: 10.1145/3531146.3533229

[27] M. Gurman. Samsung Bans Staff's AI Use After Spotting ChatGPT Data Leak, 2023.

[28] C. G. Healey. Choosing effective colours for data visualization. In *Proceedings of the IEEE Visualization Conference*, pp. 263–270. IEEE Computer Society, Los Alamitos, CA, USA, 1996. doi: 10.1109/VISUAL.1996.568118

[29] A. K. Hopkins, M. Correll, and A. Satyanarayan. VisuaLint: Sketchy in situ annotations of chart construction errors. *Computer Graphics Forum*, 39(3):219–228, 2020. doi: 10.1111/cgf.13975

[30] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo. VizML: A machine learning approach to visualization recommendation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, p. 128:1–128:12. ACM, New York, NY, USA, 2019. doi: 10.1145/3290605.3300358

[31] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023. doi: 10.1145/3571730

[32] A. Key, B. Howe, D. Perry, and C. Aragon. VizDeck: Self-organizing dashboards for visual analytics. In *Proceedings of the ACM Conference on Management of Data*, p. 681–684. ACM, New York, NY, USA, 2012. doi: 10.1145/2213836.2213931

[33] N. Lee, W. Ping, P. Xu, M. Patwary, P. N. Fung, M. Shoeybi, and B. Catanzaro. Factuality enhanced language models for open-ended text generation. In *Advances in Neural Information Processing Systems*, vol. 35, pp. 34586–34599. Curran Associates, Inc., Red Hook, NY, USA, 2022.

[34] F. Lekschas, S. Ampanavos, P. Siangliulue, H. Pfister, and K. Z. Gajos. Ask me or tell me? enhancing the effectiveness of crowdsourced design feedback. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2021. doi: 10.1145/3411764.3445507

[35] F. Liu, J. Eisenschlos, F. Piccinno, S. Krichene, C. Pang, K. Lee, M. Joshi, W. Chen, N. Collier, and Y. Altun. DePlot: One-shot visual language reasoning by plot-to-table translation. In *Proceedings of the Findings of the Association for Computational Linguistics*, pp. 10381–10399. Association for Computational Linguistics, Toronto, Canada, 2023. doi: 10.18653/v1/2023.findings-acl.660

[36] F. Liu, J. M. Eisenschlos, F. Piccinno, S. Krichene, C. Pang, K. Lee, M. Joshi, W. Chen, N. Collier, and Y. Altun. DePlot: One-shot visual language reasoning by plot-to-table translation. *CoRR*, abs/2212.10505, 2022. doi: 10.48550/arXiv.2212.10505

[37] V. Liu and L. B. Chilton. Design guidelines for prompt engineering text-to-image generative models. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 1–23. ACM, New York, NY, USA, 2022. doi: 10.1145/3491102.3501825

[38] Z. Liu, J. Thompson, A. Wilson, M. Dontcheva, J. Delorey, S. Grigg, B. Kerr, J. Stasko, and U. 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*, pp. 123:1–123:13. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3173697

[39] Y. Luo, X. Qin, N. Tang, and G. Li. DeepEye: Towards automatic data visualization. In *Proceedings of the IEEE International Conference on Data Engineering*, pp. 101–112. IEEE, Piscataway, NJ, USA, 2018. doi: 10.1109/ICDE.2018.00019

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

[41] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, 2007. doi: 10.1109/TVCG.2007.70594

[42] K. Marriott, B. Lee, M. Butler, E. Cutrell, K. Ellis, C. Goncu, M. A. Hearst, K. F. McCoy, and D. A. Szafir. Inclusive data visualization for people with disabilities: a call to action. *Interactions*, 28(3):47–51, 2021. doi: 10.1145/3457875

[43] A. McNutt, G. Kindlmann, and M. Correll. Surfacing visualization mirages. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, p. 1–16. ACM, New York, NY, USA, 2020. doi: 10.1145/3313831.3376420

[44] G. Mialon, R. Dessì, M. Lomeli, C. Nalmpantis, R. Pasunuru, R. Raileanu, B. Rozière, T. Schick, J. Dwivedi-Yu, A. Celikyilmaz, E. Grave, Y. LeCun, and T. Scialom. Augmented language models: a survey. *Transactions on Machine Learning Research*, 2023, 2023.

[45] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in Draco. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):438–448, 2019. doi: 10.1109/TVCG.2018.2865240

[46] N. Mündler, J. He, S. Jenko, and M. T. Vechev. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *CoRR*, abs/2305.15852, 2023. doi: 10.48550/arXiv.2305.15852

[47] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, Nov. 2009. doi: 10.1109/TVCG.2009.111

[48] T. Munzner. *Visualization Analysis and Design*. CRC Press, Boca Raton, FL, USA, 2014.

[49] J. Oppenlaender, E. Kuosmanen, A. Lucero, and S. Hosio. Hardhats and bungaloos: Comparing crowdsourced design feedback with peer design feedback in the classroom. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2021. doi: 10.1145/3411764.3445380

[50] J. Oppenlaender, T. Tiropanis, and S. Hosio. Crowdui: Supporting web design with the crowd. *Proceedings of the ACM on Human-Computer Interaction*, 4(EICS), 2020. doi: 10.1145/3394978

[51] D. Ren, T. Höllerer, and X. Yuan. iVisDesigner: Expressive interactive design of information visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2092–2101, 2014. doi: 10.1109/TVCG.2014.2346291

[52] D. Ren, B. Lee, and M. Brehmer. Charticulator: Interactive construction of bespoke chart layouts. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):789–799, 2019. doi: 10.1109/TVCG.2018.2865158

[53] L. Reynolds and K. McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems*, pp. 1–7. ACM, New York, NY, USA, 2021. doi: 10.1145/3411763.3451760

[54] T.-M. Rhyne. *Applying Color Theory to Digital Media and Visualization*. CRC Press, Boca Raton, FL, USA, 2016.

[55] N. Riche, M. Duvinage, M. Mancas, B. Gosselin, and T. Dutoit. Saliency and human fixations: State-of-the-art and study of comparison metrics. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1153–1160. IEEE Computer Society, Los Alamitos, CA, USA, 2013. doi: 10.1109/ICCV.2013.147

[56] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design environment. *Computer Graphics Forum*, 33(3):351–360, 2014. doi: 10.1111/cgf.12391

[57] J. Schwabish, S. J. Popkin, and A. Feng. Do no harm guide: Centering accessibility in data visualization. Urban Institute, 2022. Accessed on 24 Feb 2025.

[58] M. Sedlmair, M. D. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012. doi: 10.1109/TVCG.2012.213

[59] S. Shin. *Simulation, Representation, and Automation: Human-Centered Artificial Intelligence for Augmenting Visualization Design*. PhD thesis, University of Maryland, College Park, 2024.

[60] S. Shin, S. Chung, S. Hong, and N. Elmqvist. A Scanner Deeply: Predicting gaze heatmaps on visualizations using crowdsourced eye movement data. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):396–406, 2023. doi: 10.1109/TVCG.2022.3209472

[61] S. Shin, S. Hong, and N. Elmqvist. Perceptual Pat: A virtual human visual system for iterative visualization design. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, p. 811:1–811:17. ACM, New York, NY, USA, 2023. doi: 10.1145/3544548.3580974

[62] C. Stokes, V. Setlur, B. Cogley, A. Satyanarayan, and M. 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, 2023. doi: 10.1109/TVCG.2022.3209383

[63] D. A. Szafir. Modeling color difference for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):392–401, 2018. doi: 10.1109/TVCG.2017.2744359

[64] 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.

[65] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1983.
[66] C. Wang, I. Yeh, and H. M. Liao. You only learn one representation: Unified network for multiple tasks. *CoRR*, abs/2105.04206, 2021.
[67] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar. MATCHA: speeding up decentralized SGD via matching decomposition sampling. *CoRR*, abs/1905.09435, 2019.
[68] Y. Wang, W. Wang, A. Abdelhafez, M. Elfares, Z. Hu, M. Bâce, and A. Bulling. SalChartQA: Question-driven saliency on information visualisations. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2024. doi: 10.1145/3613904.3642942
[69] M. Wattenberg and D. Fisher. A model of multi-scale perceptual organization in information graphics. In *Proceedings of the IEEE Symposium on Information Visualization*, pp. 23–30, 2003. doi: 10.1109/INFVIS.2003.1249005
[70] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus. Emergent abilities of large language models. *CoRR*, abs/2206.07682, 2022. doi: 10.48550/arXiv.2206.07682
[71] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., Red Hook, NY, USA, 2022.
[72] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, 2016. doi: 10.1109/TVCG.2015.2467191
[73] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. D. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pp. 2648–2659. ACM, New York, NY, USA, 2017. doi: 10.1145/3025453.3025768
[74] A. Xu, S.-W. Huang, and B. Bailey. Voyant: Generating structured feedback on visual designs using a crowd of non-experts. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work & Social Computing*, p. 1433–1444. ACM, New York, NY, USA, 2014. doi: 10.1145/2531602.2531604
[75] Y.-C. G. Yen, J. O. Kim, and B. P. Bailey. Decipher: An interactive visualization tool for interpreting unstructured design feedback from multiple providers. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, p. 1–13. ACM, New York, NY, USA, 2020. doi: 10.1145/3313831.3376380
[76] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, O. Bousquet, Q. Le, and E. H. Chi. Least-to-most prompting enables complex reasoning in large language models. *CoRR*, abs/2205.10625, 2022. doi: 10.48550/arXiv.2205.10625

Sungbok Shin received the Ph.D. degree in 2024 from University of Maryland, College Park in College Park, MD, USA. He is a postdoctoral researcher at Team Aviz in Inria-Saclay and Université Paris-Saclay in Saclay, France. His research interest is in Human-Centered AI, visualization, and HCI.

Sanghyun Hong received the Ph.D. degree in 2021 from University of Maryland, College Park in College Park, MD, USA. He is an assistant professor in the Department of Computer Science at Oregon State University in Corvallis, OR, USA. His research interest is at the intersection of privacy, security and machine learning. He is the recipient of the Google Faculty Research Award 2023 and was also selected as a DARPA Riser 2022.

Niklas Elmqvist received the Ph.D. degree in 2006 from Chalmers University of Technology in Göteborg, Sweden. He is a Villum Investigator and professor in the Department of Computer Science at Aarhus University in Aarhus, Denmark. He was previously faculty at University of Maryland, College Park from 2014 to 2023, and at Purdue University from 2008 to 2014. His research interests include visualization, HCI, and human-centered AI. He is a Fellow of the IEEE and the ACM.