Exploring the Capability of LLMs in Performing Low-Level Visual Analytic Tasks on SVG Data Visualizations

Zhongzheng Xu*
Brown University

Emily Wall†
Emory University

Abstract

Data visualizations help extract insights from datasets, but reaching these insights requires decomposing high level goals into low-level analytic tasks that can be complex due to varying degrees of data literacy and visualization experience. Recent advancements in large language models (LLMs) have shown promise for lowering barriers for users to achieve tasks such as writing code and may likewise facilitate visualization insight. Scalable Vector Graphics (SVG), a text-based image format common in data visualizations, matches well with the text sequence processing of transformer-based LLMs. In this paper, we explore the capability of LLMs to perform 10 low-level visual analytic tasks defined by Amar, Eagan, and Stasko directly on SVG-based visualizations [2]. Using zero-shot prompts, we instruct the models to provide responses or modify the SVG code based on given visualizations. Our findings demonstrate that LLMs can effectively modify existing SVG visualizations for some tasks like *Cluster* but perform poorly on tasks requiring mathematical operations like *Compute Derived Value*. We also discovered that LLM performance can vary based on factors such as the number of data points, the presence of value labels, and the chart type. Our findings contribute to gauging the general capabilities of LLMs and highlight the need for further exploration and development to fully harness their potential in supporting visual analytic tasks.

Index Terms: Data Visualization, Large Language Models (LLMs), Visual Analytics Tasks, Scalable Vector Graphics (SVG).

1 Introduction

Data visualizations help people obtain meaningful insights from a dataset by encoding data into visual features such as length, position, shape, or color [19, 8]. Researchers have suggested that people often look at a chart with a high-level goal [2], such as: *get an overview of the current stock market*. These high-level goals can be deconstructed into low-level tasks; following the example, it could include tasks such as *finding the extreme values* of all-time stock prices or *determining the range* of the stock prices over the course of a day [2]. Amar, Eagan, and Stasko identified 10 such low-level visual analytic tasks that users commonly perform when exploring visualizations: *Retrieving Values, Filter, Compute Derived Values, Find Extremum, Sort, Determine Ranges, Characterize Distribution, Find Anomalies, Cluster,* and *Correlate* [2]. For users to achieve their high-level goals, it may require them to decompose it into low-level tasks, the success of which may depend on several factors such as data literacy [13] and experience levels [3, 15, 16].

Recent advances in large language models (LLMs) have empowered people to complete difficult tasks such as reviewing legal documents [9], learning new languages [21], and writing software code [23] with greater ease. In the visualization community, researchers have incorporated LLMs into Natural Language Interfaces (NLIs) for chart creation [38, 28, 18, 17, 10] and automatic chart summarization tools [22, 26]. A recent study found that LLMs are capable of understanding and modifying visualizations presented in Scalable Vector Graphics (SVG) format[7]. Unlike raster-based images, SVG is a text-based image format that uses mathematical equations to define shapes, paths, and other elements through XML code [35]. This text-based XML format of SVG images suggests that they may align well with transformer-based models' sequential text processing. Several studies have explored the capability of LLMs in understanding and manipulating SVG images [4, 39], but modifying existing visualizations is often a secondary focus, limited to editing styles such as changing background color or adding grid lines[17, 37].

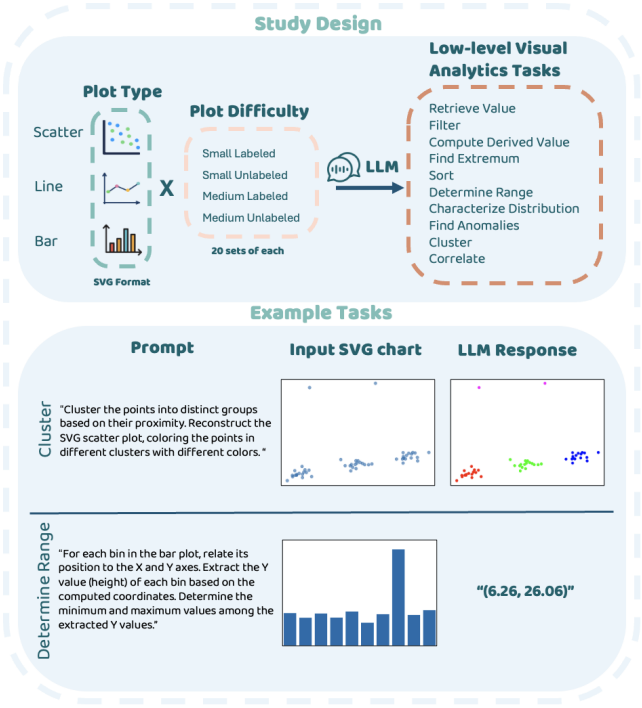

Figure 1: Illustration of the study design (top) and examples of the *Cluster* and *Determine Range* tasks (bottom).

In this paper, we explore an alternative usage of LLMs: to perform low-level visual analytic tasks directly on SVG-based visualizations. If LLMs can perform these low-level analytic tasks, it can reduce the barrier for end users to interact with data and gain insights. We thus conducted an exploratory evaluation of the ability of LLMs to understand and manipulate data visualizations using zero-shot prompts and instructed LLMs to perform tasks outlined by Amar et al. [2] on a set of generated data visualizations. We observed that the LLM can effectively modify existing SVG visualizations for specific tasks involving pattern recognition such as *Cluster* and *Find Anomalies*. However, it performed poorly on tasks requir-

*e-mail: zhongzheng_xu@brown.edu
†e-mail: emily.wall@emory.edu

ing complex mathematical operations, such as *Compute Derived Values* and *Correlation*. Our findings also highlight the LLM's capability to extract data point values directly from labeled SVG plots. We noted variations in performance based on the number of data points and the presence of value labels on the SVG charts. However, these effects were not consistent across all evaluated tasks. This baseline can provide valuable insights for future visualization researchers who are developing tools that incorporate LLMs and further contribute to gauging the general intelligence of LLMs.

2 Related Work

Advancements in LLMs have sparked interest in bridging natural language processing and data visualization. Researchers have developed NLIs that enable the creation of data visualizations through natural language prompts [38, 28, 18, 17, 10] and automatic chart summarization tools to generate captions for existing data visualizations [22, 26]. While some tools allow modifying existing visualizations, it is often limited to editing visual styles, such as colors or grid lines [17]. Many of these tools require access to the original data, and few studies have explored using LLMs for direct chart manipulations. One example is the work by Shi et al. [37], which focuses on using abstract natural language to refine color palettes in existing visualizations.

Evaluations of the capabilities of LLMs across various tasks have been conducted, including their math ability, coding skills, and machine translation, etc. [5, 42]. Several recent studies have explored the capability of LLMs in understanding and manipulating SVG images, such as changing element color in the image or editing animations [4, 39]. In data visualization, Vázquez et al. evaluated the performance of several NLIs and libraries in creating charts using natural language [40]. Another work by Chen et al. used ChatGPT to complete an introductory data visualization course, revealing that LLMs can understand and manipulate SVG visualizations [7].

However, there is a lack of research systematically evaluating LLMs' capability to perform the low-level tasks that users typically carry out when exploring and interpreting data visualizations. We aim to shape our evaluation similar to that presented by Yuan et al. [42] who evaluated LLM performance on elementary arithmetic tasks. That is, we will systematically explore LLMs' capability across low-level analytic tasks identified by Amar, Eagan, and Stasko [2].

3 Method

In this study, we aim to evaluate the capability of LLMs to execute low-level visual analytic tasks on SVG-based data visualizations. We designed an experiment that assesses 3 chart types {scatterplot, line chart, and bar chart} × 2 dataset sizes {small, medium} × 2 labeling schemas {labeled, unlabeled} × 10 visual analytic task prompts {retrieve value, filter, compute derived value, find extremum, sort, determine range, characterize distribution, find anomalies, cluster, correlate}. We provided SVG and prompts as input to an LLM and subsequently evaluated task success. We detail the full methodology in this section and report on our findings in the next section.

Visualization Selection. We evaluated the capability of LLMs to perform low-level visual analytic tasks on three chart types including scatterplots, line charts, and bar charts. These chart types are among the most widely used according to recent surveys [25, 34] and are also the commonly used in recent developments of NLIs for data visualization [27, 28, 17, 38]. Extensive research has been conducted on the effectiveness of these chart types in supporting low-level analytic tasks, such as finding clusters [33, 20], identifying correlations [30, 29], and identifying trends [1, 41]. While these chart types do not cover the full spectrum of visualizations encountered in the wild, they are well-defined, goal-specific, and suitable for this particular study.

Generation of Data and Stimuli. We generated the SVGs using Python. The data generation pipeline was designed to create datasets suitable for performing certain low-level tasks, such as *Cluster*. For scatterplots, points were initialized in multiple clusters separated by a random bias, with points within each cluster following a normal distribution and a specified x-y correlation factor. Line chart and bar chart datasets were generated by initializing points with a specified correlation factor (line chart) or random values within a range (bar chart) with random noise. Outlier points or bins were added to support the *Find Anomalies* task for all three chart types. The visualizations were plotted using the *matplotlib* library.

Our preliminary experimentation revealed that LLM's performance could vary based on the number of data points and the presence of data encoding labels. Thus, we generated two dataset sizes {small,medium} and two labeling schemas {labeled, unlabeled} for each chart type. We created 20 unique datasets for each combination to account for dataset noise. For the task *Characterize Distribution*, we generated an additional 20 datasets for each combination of chart type, dataset size, and labeling schema to evaluate the LLMs' ability to classify different distributions for scatterplots and bar charts, as the other datasets used a consistent distribution type. In total, we generated 400 unique stimuli (3 chart types × 2 dataset sizes × 2 labeling schemas × 20 random instantiations + 160 datasets for *Characterize Distribution*). Within each chart type, the number of clusters, outliers, and initialization parameters were kept consistent. For labeled charts, we used the *adjustText* library to ensure distinguishable labels [12]. Full details on data generation and accompanying code are included in supplemental materials[1].

Prompt Engineering. We employed OpenAI's *gpt-4-turbo-preview* model [32] to perform the 10 low-level tasks on our generated datasets. It has been proven that prompts significantly impact the quality and correctness of the output of LLMs [24, 14]; therefore, after preliminary experimentation, we designed our prompt to follow the format: <Low-level task description><SVG code>. In task descriptions, we instruct the model that it will answer questions or perform modification on a provided SVG code. Specifically, we followed the techniques and principles proposed in [36, 31, 6] and explicitly defined the inputs and outputs of our tasks, as well as itemizing them into small steps. We then appended chart SVG code directly after the task descriptions. An example of the prompts used in our study is shown in Figure 2 for the *Find Anomalies* task.

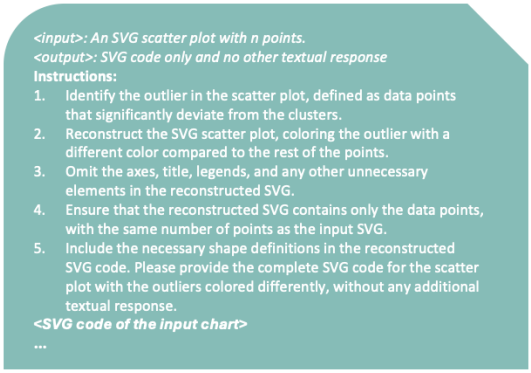

Figure 2: An example of the prompt for the task *Find Anomalies*.

Measuring Task Success. To provide clarity on the specific tasks we evaluated, we present the tasks that the LLM performed below and the metric used to assess performance.

[1] https://github.com/lebretou/SVG_taxonomy

Some low-level visual analytic tasks, such as *Compute Derived Value*, are categories that encompass multiple specific tasks [2]. For the purpose of our study, we selected one representative task for each category that aligns with the definition (e.g., for *Compute Derived Value*, we chose to compute the mean only).

- *Retrieve Value*: Extract a list of coordinates (scatterplot, line chart) or bar heights (bar chart).
- *Filter*: Create a plot with a subset of points (scatterplot, line chart) or bars (bar chart) that satisfy a given filtering criterion
- *Compute Derived Value*: Calculate the mean across the Y-axis (scatterplot, line chart, bar chart)
- *Find Extremum*: Highlight the points (scatterplot, line chart) or bars (bar chart) with minimum and maximum values in a different color
- *Sort*: Rearrange the bars (bar chart) in descending order based on their values
- *Determine Range*: Identify the range of values across the Y-axis (scatterplot, line chart, bar chart)
- *Characterize Distribution*: Identify the type of distribution exhibited by the points (scatterplot) or bins (bar chart)
- *Find Anomalies*: Highlight anomalous points (scatterplot, line chart) or bars (bar chart) in a different color
- *Cluster*: Group points (scatterplot) into clusters, each represented by a different color
- *Correlate*: Add a fitted line that linearly fits the data points (scatterplot, line chart) in the plot

To measure the success of each task, we generated ground truth as answer keys for all the tasks. Although more complex evaluation metrics could be employed to measure partial success, we selected Exact Match (EM) [5] as our primary evaluation metric.

4 Result

In this section, we present the results of the LLM's performance on each low-level visual analytic task using the Exact Match metric. Table 1 provides an overview of the scores across different chart types and dataset size and labeling schemas. We structure this section by discussing the results for each low-level task individually, comparing the performance across scatterplots, line charts, and bar charts. Note that many task success rates will fall in increments of 5% given that we measure exact match across 20 random dataset instantiations, with the exception of the *retrieve value* task, for which we compute the exact match for each of the N respective data points.

Task	Difficulty	scatterplot	line chart	bar chart
Retrieve Value	small labeled	99.1%	100%	100%
	small unlabeled	0%	0%	0%
	medium labeled	86.4%	100%	100%
	medium unlabeled	0%	0%	0%
Filter	small labeled	0%	55%	35%
	small unlabeled	0%	40%	20%
	medium labeled	0%	60%	5%
	medium unlabeled	0%	65%	0%
Compute Derived Value	small labeled	0%	0%	0%
	small unlabeled	0%	0%	0%
	medium labeled	0%	0%	0%
	medium unlabeled	0%	0%	0%
Find Extremum	small labeled	35%	70%	5%
	small unlabeled	55%	80%	10%
	medium labeled	35%	90%	30%
	medium unlabeled	35%	85%	30%
Sort	small labeled	N/A	N/A	20%
	small unlabeled	N/A	N/A	15%
	medium labeled	N/A	N/A	5%
	medium unlabeled	N/A	N/A	0%
Determine Range	small labeled	55%	95%	0%
	small unlabeled	0%	0%	0%
	medium labeled	5%	15%	0%
	medium unlabeled	0%	0%	0%
Characterize Distribution	small labeled	0%	N/A	0%
	small unlabeled	0%	N/A	0%
	medium labeled	0%	N/A	0%
	medium unlabeled	0%	N/A	0%
Find Anomalies	small labeled	80%	100%	85%
	small unlabeled	60%	95%	95%
	medium labeled	65%	95%	70%
	medium unlabeled	45%	100%	75%
Cluster	small labeled	95%	N/A	N/A
	small unlabeled	95%	N/A	N/A
	medium labeled	95%	N/A	N/A
	medium unlabeled	95%	N/A	N/A
Correlate	small labeled	0%	0%	N/A
	small unlabeled	0%	0%	N/A
	medium labeled	0%	0%	N/A
	medium unlabeled	0%	0%	N/A

Table 1: The performance of the LLM on low-level visual analytic tasks across different chart types and difficulty levels. The percentage scores indicate the Exact Match scores between the LLM's output and the ground truth answer key. "N/A" indicates that the task was not tested for a particular chart type due to its inapplicability.

1. Retrieve Value: The LLM achieved 100% accuracy for line charts and bar charts with labeled data values. For scatterplots, the accuracy was slightly lower due to missed and incorrectly identified data points in some responses. When charts lacked data value labels, the LLM couldn't retrieve exact values for all three chart types, with responses often containing imaginary data points. Scatterplots were the most affected. However, for many line chart and bar chart responses, the retrieved points, when plotted, resulted in shapes similar to the original data points. Pearson correlation revealed that the LLM retrieved points resulting in the same shape as the original chart in 17/20 "small" and 10/20 "medium" cases for line charts, and 15/20 "small" and 7/20 "medium" cases for bar charts, achieving a 99% correlation factor. The remaining responses were hallucinations with no correlation to the original data. Inspection revealed that the LLM extracted x,y coordinates from the SVG code's point position definitions, despite the prompt stating not to do so (as SVG pixel coordinates don't equate to point positions on the defined scales). Figure 3 compares the retrieved data points, extracted directly from SVG position definitions, against the original data points.

2. Filter: The LLM had the highest accuracy for line charts, followed by bar charts, with no successful cases for scatterplots. For line charts, accuracy was higher with more data points ("medium" cases) and unlabeled data points.

Scatterplots suffered from hallucination, with the LLM returning a subset of points that didn't meet the specified filtering criteria or sometimes including all original points without filtering. Some incorrect responses for line charts and bar charts returned a subset of original points that didn't meet the desired filtering conditions.

3. Compute Derived Value: The LLM had no success, regardless of whether the points were labeled with their exact values or not. The returned values were generally hallucinations. We discuss this finding in greater detail in Section 5.

4. Find Extremum: The LLM's accuracy was highest for line charts, followed by scatterplots and bar charts. Interestingly, in some cases, the LLM performed better when data values were not labeled. For line and bar charts, accuracy improved with more data points. In several incorrect responses across all chart types, the LLM highlighted only one extremum or sometimes local extrema instead of global extrema.

5. Sort: The LLM achieved the highest accuracy for bar charts in the "small labeled" cases, followed by "small unlabeled" and "medium labeled" cases, but failed to correctly sort bars for "medium unlabeled" cases. Inspection of failed responses revealed that the LLM attempted to rearrange the bars, but not always in the specified "descending order," sometimes outputting bars in ascending order instead.

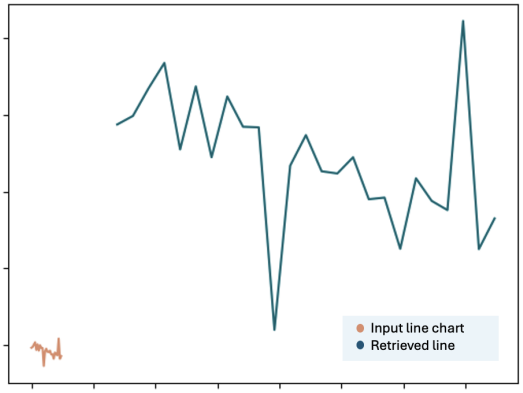

Figure 3: Comparison of retrieved data points (teal) extracted by the LLM directly from SVG position definitions and the original data points (orange).

6. Determine Range: The LLM had the best accuracy for line charts, followed by scatterplots, and it was not able to determine the range for bar charts. When the data points were not labeled, the LLM failed to determine the range for all three chart types. As with the *Retrieve Value* task, the LLM gave imaginary points and mistakenly extracted (x,y) point positions defined in the SVG code.

7. Characterize Distribution: The LLM had no success for scatterplots or bar charts. The data were generated according to normal, binomial, or uniform distributions. The LLM responded with hallucinations like "positive correlation" for all 20 plots. However, when the correct distribution options were explicitly included in the prompt (e.g., "The answer should be one of [normal, uniform, binomial]."), the LLM's accuracy improved to 37.5% for scatterplots and 35% for bar charts, reflecting chance accuracy.

8. Find Anomalies: The LLM had the highest accuracy for line charts, followed by bar charts and scatterplots. Notably, for line charts and bar charts, the performance was comparable when data values were labeled and not labeled, though worse when not labeled for scatterplots. In the failed responses, the LLM sometimes highlighted incorrect points, such as a point among a cluster, and sometimes it would only highlight one outlier instead of two for the "medium" cases.

9. Cluster: The LLM successfully highlighted different clusters in colors with an accuracy of 95% for all cases. In the failed responses, the LLM provided incorrect clustering results with more clusters than the number specified in the prompt.

10. Correlate: The LLM failed to add a properly fitted line to the chart for both scatterplots and line charts. Instead, most of the responses included an arbitrarily placed line, indicating hallucination. Notably, this was the only task for which the LLM refused to respond with an SVG code output, stating, "I am unable to modify SVG files."

5 Discussion and Conclusion

In this section, we discuss what we expected and did not expect from the performance of the LLM in performing low-level visual analytic tasks, including analysis of factors that influenced the LLM's performance. We synthesize the implications of our findings for the field of data visualization, then identify limitations of our approach and future research directions.

The Expected. The LLM performed poorly when complex mathematical operations were required. For example, retrieving unlabeled data point values requires subtraction and division based on axis labels. Similarly, computing the mean across an axis involves addition, counting, and division. This aligns with previous research on LLMs' limited capability in handling long expressions, especially when directly outputting answers [42]. However, there is promise in integrating scripting approaches alongside LLMs, as ChatGPT does with its web interface (though not in the API), discussed further in Implications.

The Unexpected. Contrary to expectations, fewer data points did not always lead to higher accuracy. For tasks like *Finding Extremum*, the LLM had higher accuracy for line and bar charts with more data points. Likewise, data point labels do not always affect performance; in *Finding Extremum*, the LLM performed comparably regardless of labeling, indicating its ability to extract information from SVG shape definitions alone. Surprisingly, despite accurately retrieving values from labeled charts, the LLM's performance in determining value ranges was poor. However, the LLM's high accuracy in clustering scatterplot points suggests its adeptness in recognizing and grouping patterns.

Implications. The differences in LLMs' capability to perform low-level visual analytic tasks offer valuable insights. Unlike previous tools like NLIs that require data to create plots, LLMs can directly perform low-level tasks on existing visualizations without the original data. However, relying solely on LLMs' inherent abilities may not suffice, as they can "lose track" of themselves in intermediate steps [31, 6]. Future research could explore bolstering training of sub-tasks where LLMs fail to perform accurately, as fine-tuning LLMs has been shown to substantially enhance their abilities in trained domains, suggesting the possibility of developing reliable tools that incorporate LLMs for tasks like clustering points in a scatterplot [11, 7].

Combining LLMs with other techniques could likewise enhance their effectiveness and accuracy in low-level visual analytic tasks. ChatGPT's web interface, which supports running code within the session, demonstrates how integrating LLMs with other tools can improve accuracy in tasks like *Determine Range* by executing Python code on coordinates extracted from data labels [32]. Finally, our motivation began with the hope that LLMs could facilitate end-users accomplishing high-level goals of data comprehension and insight generation from data visualizations. For these findings to be put into practice, we must understand how to compose the low-level tasks that have high accuracy into high-level visual analytic goals, drawing inspiration from the decomposition approach attempted by Tian et al. in their NLI for chart creation [38].

Limitations and Future Work.
Our work is limited by the small and homogenous set of charts evaluated. Future studies could test a wider array of chart types and variations. Additionally, our evaluation used only OpenAI's *gpt-4-turbo-preview* [32]. Future research could compare the performance across different models. Moreover, we utilized zero-shot prompts, depending solely on the model's built-in knowledge. As [7] indicates, in-context learning and fine-tuning might enhance LLM performance on specialized tasks.

Conclusion. We conducted an exploratory evaluation of an LLM's potential in performing low-level visual analytic tasks on SVG data visualizations. LLMs are capable of executing tasks like identifying outliers and clustering points in scatterplots, but struggle with tasks requiring complex mathematical operations. Performance varies based on the number of data points, value label presence, and chart type. Our findings contribute to the growing research on LLMs in data visualization and highlight the need for further exploration and development to harness their potential in supporting visual analytic tasks.

Acknowledgments

This work was partially supported by NSF award IIS-2311574.

REFERENCES

[1] D. Albers, M. Correll, and M. Gleicher. Task-driven evaluation of aggregation in time series visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 551–560, 2014. 2

[2] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 111–117. IEEE, 2005. 1, 2, 3

[3] A. Burns, C. Lee, R. Chawla, E. Peck, and N. Mahyar. Who do we mean when we talk about visualization novices? In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2023. 1

[4] M. Cai, Z. Huang, Y. Li, H. Wang, and Y. J. Lee. Leveraging large language models for scalable vector graphics-driven image understanding. *arXiv preprint arXiv:2306.06094*, 2023. 1, 2

[5] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2023. 2, 3

[6] B. Chen, Z. Zhang, N. Langrené, and S. Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735*, 2023. 2, 4

[7] Z. Chen, C. Zhang, Q. Wang, J. Troidl, S. Warchol, J. Beyer, N. Gehlenborg, and H. Pfister. Beyond generating code: Evaluating gpt on a data visualization course. In *2023 IEEE VIS Workshop on Visualization Education, Literacy, and Activities (EduVis)*, pp. 16–21. IEEE, 2023. 1, 2, 4

[8] M. Dastani. The role of visual perception in data visualization. *Journal of Visual Languages & Computing*, 13(6):601–622, 2002. 1

[9] A. Deroy, K. Ghosh, and S. Ghosh. How ready are pre-trained abstractive models and llms for legal case judgement summarization? *arXiv preprint arXiv:2306.01248*, 2023. 1

[10] V. Dibia. Lida: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models. *arXiv preprint arXiv:2303.02927*, 2023. 1, 2

[11] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023. 4

[12] I. Flyamer, Z. Xue, A. Li, J. L. Espinoza, N. Morshed, V. Vazquez, R. Neff, et al. Phlya/adjusttext: 0.8. *Zenodo*, 2023. 2

[13] S. Franconeri, L. Padilla, P. Shah, J. Zacks, and J. Hullman. The science of visual data communication: What works (vol 22, pg 110, 2021). *PSYCHOLOGICAL SCIENCE IN THE PUBLIC INTEREST*, 23(1):41–42, 2022. 1

[14] H. Gonen, S. Iyer, T. Blevins, N. A. Smith, and L. Zettlemoyer. Demystifying prompts in language models via perplexity estimation, 2022. 2

[15] L. Grammel, M. Tory, and M.-A. Storey. How information visualization novices construct visualizations. *IEEE transactions on visualization and computer graphics*, 16(6):943–952, 2010. 1

[16] S. Gupta, A. Karduni, and E. Wall. Belief decay or persistence? a mixed-method study on belief movement over time. In *Computer Graphics Forum*, vol. 42, pp. 111–122. Wiley Online Library, 2023. 1

[17] Y. Han, C. Zhang, X. Chen, X. Yang, Z. Wang, G. Yu, B. Fu, and H. Zhang. Chartllama: A multimodal llm for chart understanding and generation. *arXiv preprint arXiv:2311.16483*, 2023. 1, 2

[18] M.-H. Hong and A. Crisan. Conversational ai threads for visualizing multidimensional datasets. *arXiv preprint arXiv:2311.05590*, 2023. 1, 2

[19] M. Islam and S. Jin. An overview of data visualization. In *2019 International Conference on Information Science and Communications Technologies (ICISCT)*, pp. 1–7. IEEE, 2019. 1

[20] H. Jeon, G. J. Quadri, H. Lee, P. Rosen, D. A. Szafir, and J. Seo. Clams: a cluster ambiguity measure for estimating perceptual variability in visual clustering. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 2

[21] J. Jeon and S. Lee. Large language models in education: A focus on the complementary relationship between human teachers and chatgpt. *Education and Information Technologies*, 28(12):15873–15892, 2023. 1

[22] S. Kantharaj, R. T. K. Leong, X. Lin, A. Masry, M. Thakkar, E. Hoque, and S. Joty. Chart-to-text: A large-scale benchmark for chart summarization. *arXiv preprint arXiv:2203.06486*, 2022. 1, 2

[23] M. Kazemitabaar, X. Hou, A. Henley, B. J. Ericson, D. Weintrop, and T. Grossman. How novices use llm-based code generators to solve cs1 coding tasks in a self-paced learning environment. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*, pp. 1–12, 2023. 1

[24] C. Kervadec, F. Franzon, and M. Baroni. Unnatural language processing: How do language models handle machine-generated prompts?, 2023. 2

[25] Q. Li and Q. Li. Overview of data visualization. *Embodying data: Chinese aesthetics, interactive visualization and gaming technologies*, pp. 17–47, 2020. 2

[26] A. Liew and K. Mueller. Using large language models to generate engaging captions for data visualizations. *arXiv preprint arXiv:2212.14047*, 2022. 1, 2

[27] Y. Luo, J. Tang, and G. Li. nvbench: A large-scale synthesized dataset for cross-domain natural language to visualization task. *arXiv preprint arXiv:2112.12926*, 2021. 2

[28] P. Maddigan and T. Susnjak. Chat2vis: Generating data visualisations via natural language using chatgpt, codex and gpt-3 large language models. *Ieee Access*, 2023. 1, 2

[29] D. Makowski, M. S. Ben-Shachar, I. Patil, and D. Lüdecke. Methods and algorithms for correlation analysis in r. *Journal of Open Source Software*, 5(51):2306, 2020. 2

[30] J. P. Miller, R. Taori, A. Raghunathan, S. Sagawa, P. W. Koh, V. Shankar, P. Liang, Y. Carmon, and L. Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International conference on machine learning*, pp. 7721–7735. PMLR, 2021. 2

[31] S. Mishra, D. Khashabi, C. Baral, Y. Choi, and H. Hajishirzi. Reframing instructional prompts to gptk's language. *arXiv preprint arXiv:2109.07830*, 2021. 2, 4

[32] OpenAI. Chatgpt-4: Optimizing language models for dialogue. `https://openai.com/chatgpt-4`, 2023. Accessed: 2024-04-05. 2, 4

[33] G. J. Quadri, J. A. Nieves, B. M. Wiernik, and P. Rosen. Automatic scatterplot design optimization for clustering identification. *IEEE Transactions on Visualization and Computer Graphics*, 2022. 2

[34] G. J. Quadri and P. Rosen. A survey of perception-based visualization studies by task. *IEEE transactions on visualization and computer graphics*, 28(12):5026–5048, 2021. 2

[35] A. Quint. Scalable vector graphics. *IEEE MultiMedia*, 10(3):99–102, 2003. 1

[36] L. Reynolds and K. McDonell. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–7, 2021. 2

[37] C. Shi, W. Cui, C. Liu, C. Zheng, H. Zhang, Q. Luo, and X. Ma. Nl2color: Refining color palettes for charts with natural language. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 1, 2

[38] Y. Tian, W. Cui, D. Deng, X. Yi, Y. Yang, H. Zhang, and Y. Wu. Chartgpt: Leveraging llms to generate charts from abstract natural language. *IEEE Transactions on Visualization and Computer Graphics*, 2024. 1, 2, 4

[39] T. Tseng, R. Cheng, and J. Nichols. Keyframer: Empowering animation design using large language models. *arXiv preprint arXiv:2402.06071*, 2024. 1, 2

[40] P.-P. Vázquez. Are llms ready for visualization?, 2024. 2

[41] Y. Wang, F. Han, L. Zhu, O. Deussen, and B. Chen. Line graph or scatter plot? automatic selection of methods for visualizing trends in time series. *IEEE transactions on visualization and computer graphics*, 24(2):1141–1154, 2017. 2

[42] Z. Yuan, H. Yuan, C. Tan, W. Wang, and S. Huang. How well do large language models perform in arithmetic tasks?, 2023. 2, 4