A Comparative Study of Prompt Engineering Strategies for Data Visualization Scenarios

Camile Maria Cunha Aguiar[1(✉)], Walbert Cunha Monteiro[2], Thiago Augusto Soares de Sousa[1], Roberto Yuri da Silva Franco[3], and Bianchi Serique Meiguins[1]

[1] Federal University of Pará, Pará, Brazil
{camile.aguiar,thiago.sousa}@icen.ufpa.br, bianchi@ufpa.br
[2] Federal Institute of Pará, Pará, Brazil
walbert.monteiro@ifpa.edu.br
[3] Federal Rural University of the Amazon, Pará, Brazil
roberto.franco@ufra.edu.br

Abstract. The creation of data visualizations using Large Language Models (LLMs) depends on the quality of user prompts, which may present ambiguities when not explicitly structured. In this context, this work investigates, through quantitative analysis, how four different prompt engineering strategies: no prompt engineering, few-shot prompting, Visualization Query Language (VQL), and a hybrid approach (few-shot and VQL), influence the process of prompt construction and quality in visualization generation. A repeated-measures experiment was conducted with 16 participants to compare the prompt engineering strategies. The analysis considered efficiency, structure, completion time, prompt length, and prompt complexity metrics. The results show that although no significant differences were observed in average performance, structured approaches reduce variability and increase consistency across participants. In particular, the hybrid approach achieved the best overall performance (score = 0.858), combining higher structural completeness with lower ambiguity. These findings indicate that prompt-based effectiveness is strongly associated with the explicit representation of task structure rather than any single strategy alone, highlighting the structured support mechanisms in LLM-based systems.

Keywords: Prompt Engineering · Data Visualization · Large Language Models · User evaluation · Variability · Ambiguity

1 Introduction

The use of generative Artificial Intelligence (AI) has evolved beyond text production, also supporting the creation of visualizations [20,23]. In this context, specifying user intentions through prompts becomes an important factor for the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16767, pp. 544–560, 2027.
https://doi.org/10.1007/978-3-032-30491-9_34

quality of the created visualizations [3,16,22]. In systems based on Large Language Models (LLMs), prompts are the main mechanism of interaction between the user and the model, directly influencing task and design decisions [11,21,22].

However, the interpretation of user-provided instructions may be ambiguous, especially when the task objective is not explicitly defined or when visualization requirements are not clearly specified [18]. As a result, the quality of the created visualizations may vary significantly. Although prompt engineering strategies have been proposed to mitigate this issue, there remains a gap in understanding how different approaches systematically affect prompt construction and the outcomes in data visualization tasks [8].

This work investigates the design of prompt engineering for generating data visualizations with LLMs. We evaluate how four prompt engineering approaches, prompt without engineering, prompts with few examples, Visualization Query Language (VQL), and a hybrid method (Few-shot and VQL), influence user efficiency, instruction structure, and quality of user-generated instructions, as well as the performance in the visualization generation process.

To achieve this objective, we organized our study into three core axes. The first axis, instruction structure, examines prompt elements and completeness using specific metrics. In the second construction process, we analyze user effort by measuring completion time and instruction length. The third, performance and variability, we investigate how prompt strategies differ across various visualization types and analytical tasks.

We explore four research questions: (RQ1) how do prompt engineering strategies influence efficiency; (RQ2) how do these strategies affect the structure, completeness, and complexity; (RQ3) how do approaches impact variability and consistency in participant performance; and (RQ4) how do visualization type and the nature of the analytical task influence the prompt construction process and performance.

This work offers two levels of contribution. Level one proposes a set of quantitative metrics to evaluate the prompt construction process in data visualization tasks, including completion time, prompt length, and prompt complexity, and defines a structured codebook for content analysis. Level two presents a controlled study comparing different prompt engineering strategies and provides evidence of their effects on user efficiency, consistency, and performance variability.

The remainder of this paper is organized as follows: Sect. 2 presents the concepts and prompt engineering approaches relevant to this study; Sect. 3 discusses related work; Sect. 4 describes the experiment and the adopted methodology; Sect. 5 presents the results and discussion; Sect. 6 provides final remarks; and Sect. 7 outlines the limitations and provides directions for future work.

2 Prompt Engineering Approaches

Prompt engineering has been investigated as a mechanism for guiding Large Language Models (LLMs) to perform specific tasks. Different strategies can be

used to structure the instructions provided to the model, ranging from direct approaches based solely on textual instructions to more structured methods that include examples or formal task representations. Among the approaches explored in the literature [3,17,21], the most prominent are one-shot prompting, few-shot prompting, chain-of-thought prompting, and the use of structured query languages for data visualization (VQL).

One-Shot Prompting. Involves giving the model a single instruction along with an example showing how to perform the task. This can enhance the quality of the model's responses relative to instructional prompts (zero-shot), particularly in tasks that necessitate specific formatting or output structures [3]

Few-Shot Prompting. This strategy consists of providing multiple examples as part of the model input to condition the response, thereby aligning it more closely with the task context [7]. Recent studies [7,19,20] indicate that more consistent results can be achieved when five examples are used.

Chain-of-Thought (CoT). This strategy encourages the model to explicitly generate intermediate reasoning steps before producing the final answer [16]. Studies show that CoT can improve performance in tasks involving logical reasoning and problem decomposition [9]. Although it is not the main focus of this work, this approach is relevant within the broader context of prompt engineering.

Visualization Query Language (VQL). This approach uses formal structures inspired by query languages, such as SQL, to explicitly describe visualization components [21]. In some systems, these specifications can be represented in structured formats such as JSON. In this context, the prompt is organized into instructions that define different aspects of the visualization, such as: **VISUALIZE** (type of technique), **SELECT** (variables of interest), **FROM** and **WHERE** (data source and filtering), and **GROUP BY** and **ORDER BY** (aggregation and sorting operations).

3 Related Work

This section presents studies that explore the use of prompt engineering strategies to generate and evaluate data visualizations using LLMs.

In the context of visualization generation, the work of [19] proposes LLM4Vis, an automated recommendation approach that uses few-shot prompting to leverage models' reasoning capabilities and overcome the limitations of traditional methods. Similarly, [21] investigates the use of LLMs in the NL2Vis scenario, evaluating how natural language queries can be transformed into visualizations through structured languages such as VQL.

Likewise, [24] presents LightVA, an agent-based framework that integrates multiple prompt engineering strategies, including structured templates, few-shot prompting, and self-evaluation mechanisms, focusing on visual analytics tasks. Additionally, [16] combines fine-tuning with structured logic techniques, such as Chain-of-Thought (CoT), to refine command interpretation and the creation of visualizations.

Other works investigate visualization evaluation. [5] proposes a Visual Question Response (VQA)-based framework for the automatic evaluation of visualizations created by LLMs, considering aspects of comprehension and communicative effectiveness. In contrast, [7] analyzes the use of ChatGPT as an assistant to evaluate visualization design, comparing its responses with human feedback.

Despite these advancements, existing studies primarily focus on improving model performance or automating the generation and evaluation of visualizations. However, limited attention has been given to understanding how different prompt engineering strategies influence prompt construction processes in user studies, particularly in visualization tasks.

To address this gap, this work investigates how user prompts are shaped by different prompt engineering strategies in the context of data visualization. This perspective complements existing research by focusing on the comparative evaluation of prompt engineering strategies.

4 Experiments

This section describes the experimental design, study configurations, participants, dataset, and analysis methods adopted to investigate the impact of different prompt engineering strategies on the construction of data visualizations using LLMs.

4.1 Methodology

This work adopts a repeated-measures experimental design, divided into two stages that involve data-visualization construction tasks supported by the ChatGPT Go 5.2 tool.

In the first stage of the experiment, participants were introduced to the experimental sessions, the Informed Consent Form (ICF), and the presence of a supervisor throughout the procedure, who would clarify any doubts and provide guidance at any time. Subsequently, participants received detailed instructions on the experiment's objectives, the dataset used, the adopted methodology, and the expected results.

All instructions, interactions, and prompt construction were conducted in Portuguese to ensure linguistic consistency across participants. Participants were also informed that there was no time limit for completing the experiments. Finally, they signed the ICF, formally agreeing to participate in the study.

The second stage consisted of participants performing data visualization construction and analysis tasks using the ChatGPT Go 5.2 tool. This stage was structured into four experimental sessions corresponding to the following conditions: (i) no prompt engineering, in which participants freely described prompts to construct the visualizations; (ii) few-shot prompt engineering, in which the tool was configured with 15 predefined visual examples as references; (iii) VQL-based engineering, in which participants specified prompts using a formal structure inspired by query languages; and (iv) hybrid engineering (few-shot + VQL),

which combined visual reference examples with a formal specification structure. The order of the four conditions was defined using a balanced randomization for each participant.

Sessions could be completed either continuously or at different times, depending on participant availability. For each new experimental condition, the environment configuration was previously adjusted by the experiment supervisor. Completion of each experimental condition occurred when the participant considered the result obtained through the applied prompts satisfactory relative to the expected outcomes. The second stage was completed once all four experimental conditions had been performed.

All configurations and expected results for the proposed experimental conditions are available in [1] and are further described in Sect. 4.2.

4.2 Experimental Configurations

To support the experimental conditions, four main elements were defined: (i) a dataset, (ii) visual configuration files, (iii) configuration prompts, organized into two experimental configuration sets, and (iv) a set of six reference images.

Two sets of experimental configurations were defined. The first set consists of a dataset and a configuration file, named *visualconfig*, which specifies the experiment's visual parameters, such as the color palette, typography, and visualization area dimensions (1200 × 800 pixels). This set was used in conditions that did not involve visual examples as references, specifically, the prompt without engineering and VQL-based approaches.

The second set consists of the dataset, the *visualconfig* file, and 15 additional reference images, organized into three visualization techniques, with five examples per technique. This set was used in conditions adopting the few-shot approach and the hybrid condition (few-shot + VQL).

In addition, base prompts were defined to configure the experimental environment across all conditions. These prompts specified that visualizations should be implemented in Python using the Matplotlib library, based on the provided dataset, while respecting the parameters defined in the *visualconfig* file. Furthermore, they instructed ChatGPT Go to exclusively use the information provided in the experiment, disregarding any context from previous interactions.

In the conditions adopting the few-shot approach, the prompts explicitly indicated the presence of visual reference examples, used to guide the structure and format of the generated visualizations. In contrast, in the VQL-based conditions, the prompts incorporated a formal specification structure, enabling the explicit description of visualization elements.

The reference images used as expected outputs for the three visualization techniques explored are presented in Fig. 1, which includes bar charts, line charts, and scatterplots, with two images for each technique corresponding to value comparison and extreme value identification tasks. All images were generated using ChatGPT Go 5.2. In Fig. 1 (A) value comparison by sector (CNAE) and Fig. 1 (B) distribution of operation value by BNDES sector, Fig. 1 (C) temporal evolution of operation and disbursement values and Fig. 1 (D) variation of

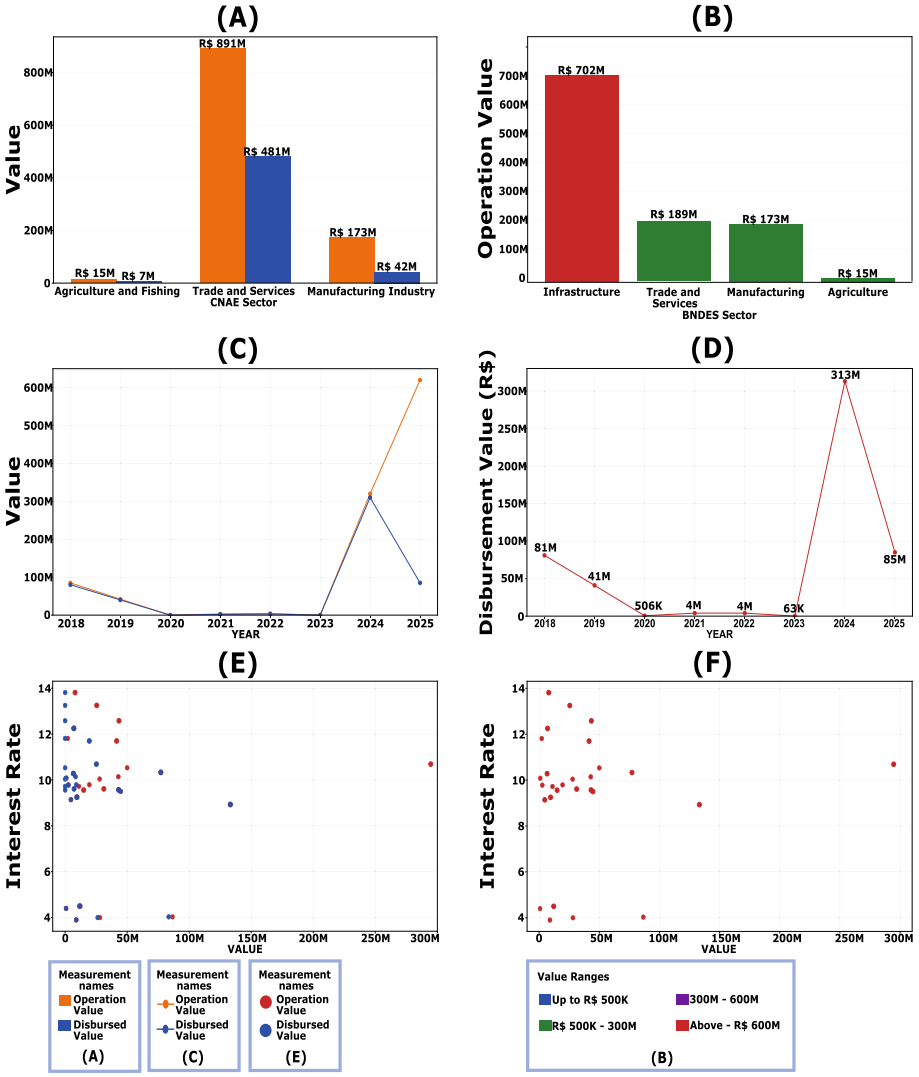

Fig. 1. Visualizations used as references during the experiment.

disbursement value over time, Fig. 1 (E) relationship between value and interest rate considering operation and disbursement values, and Fig. 1 (F) relationship between operation value and interest rate.

4.3 Participants

A total of 16 volunteers (10 male and 6 female) participated in the study, with ages ranging from 18 to 55 years (Mean = 30.5, SD = 8.9). All participants were

undergraduate or graduate students who volunteered for the study. The number of participants was defined based on experimental studies in the fields of Human–Computer Interaction and Information Visualization [13–15].

4.4 Dataset

The dataset used in this study is based on operations of the Climate Fund, which finances projects related to sustainable development, climate change, and environmental protection [2]. The dataset was released in the context of the 2025 United Nations Climate Change Conference (COP30).

The dataset contains 394 records and 19 dimensions, including 11 categorical variables (state, month, CNAE sector, client size, municipality, CNAE subsector, grouped CNAE subsector, BNDES sector, BNDES subsector, contract date, and financial institution) and 8 numerical variables (operation ID, year, municipality code, operation value, disbursed amount, interest rate, amortization period, and grace period).

4.5 Quantitative Analysis

The quantitative analysis used descriptive statistics to characterize participants' behavior in terms of effort, structure, and efficiency in prompt construction. We considered three dependent variables: Completion Time, Prompt Length, and Prompt Complexity.

Completion Time represents the total time taken to complete each task, measured from the start to the completion of prompt creation. This variable estimates the time effort involved in the prompt construction process.

Prompt Length corresponds to the size of the prompt relative to the total number of tokens, calculated using the platform [12]. This metric allows the analysis of the level of detail in the instructions and its relationship with both effort and process efficiency.

Prompt Complexity evaluates the level of instruction structuring based on the Directed Content Analysis methodology [6]. For this analysis, a category scheme was developed to reflect the fundamental requirements for proper data visualization specification, including: (i) chart type, (ii) axis mapping (x or y), (iii) aggregation operations, (iv) analytical task, (v) informative elements (titles, legends, and labels), and (vi) data formatting.

Based on these categories, we constructed a codebook with operational definitions and representative examples. The codebook definitions are presented in Table 1.

Each category was coded according to its presence or absence in the instructions, enabling the calculation of quantitative scores for structural completeness. This enabled a comparison of the level of specification between the prompts created by each participant.

Table 1. Codebook used for prompt content analysis

Category	Definition	Example
Chart type specification	Explicit mention of the type of visualization to be generated by the system.	"I want a grouped bar chart."
Axis mapping	Explicit association between dataset variables and the axes.	"The X-axis represents the year and the Y-axis represents the amount."
Data aggregation operations	Indication of aggregation operations, such as sum or average.	"Perform the summation for the BNDES sector."
Analytical task	Specification of the analytical operation, such as comparison or identification of extremes.	"Create a comparison line chart."
Informational elements	Inclusion of descriptive elements such as title, legend, axis labels, or annotations.	"Add a title and legend to the chart."
Data formatting	Specification of how data values should be presented, such as scientific notation, full values, or abbreviations.	"Use abbreviations such as 'K' for thousands and 'M' for millions."

A second evaluator conducted an independent analysis to ensure reliability. The level of agreement achieved was **83%** ($P_o = 0.830$), with a Cohen's Kappa coefficient of $\kappa = 0.71$, indicating substantial agreement [4].

In addition to the structural dimensions considered in the Prompt Complexity metric, two additional categories related to the semantic quality of instructions were included: ambiguity and redundancy. Ambiguity refers to vague or incomplete instructions that fail to clearly specify essential visualization elements. Redundancy corresponds to the unnecessary repetition of information or overly verbose descriptions that do not contribute to the structural clarity of the prompt.

5 Results and Discussion

5.1 RQ1: Efficiency in the Prompt Construction Process

To answer the first research question (RQ1), we analyzed efficiency using two variables: Completion Time and Instruction Length. To the first variable, the distribution for experimental condition, is shown in Fig. 2(A) distribution of completion time by condition, Fig. 2(B) completion time distribution with individual data points, Fig. 2(C) completion time by condition and visualization technique, and Fig. 2(D) completion time by condition and task type.

The analysis of mean completion times by condition, Fig. 2(A), was similar across the four conditions analyzed, ranging from 307.27 s (VQL) to 370.95 s

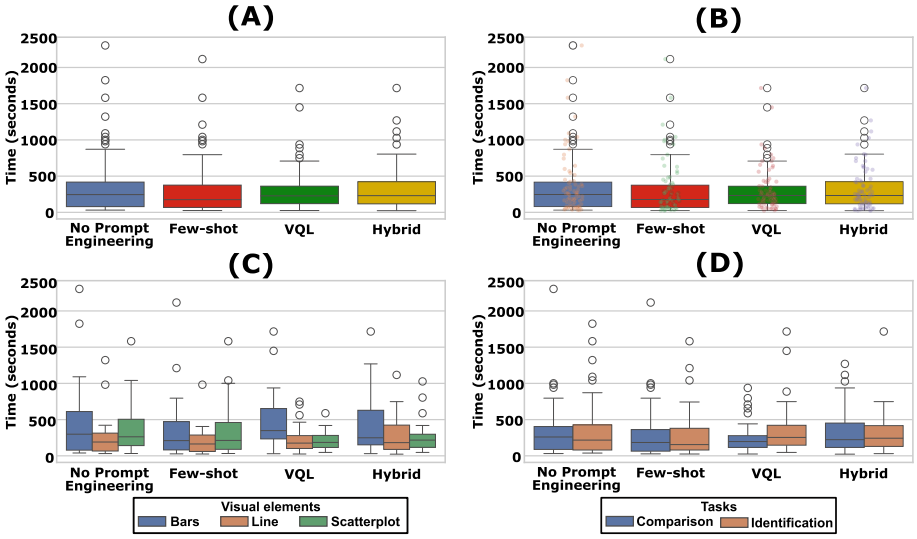

Fig. 2. Distribution of completion time variable by experimental condition.

(few-shot). The Kruskal-Wallis test was applied to evaluate the medians of the groups and did not identify statistically significant differences ($H = 1.88$,p $= 0.597$). This result suggests that the choice of engineering strategy does not significantly affect the mean time to develop instructions.

For the distribution of a participant's completion time, Fig. 2(B), the values are concentrated up to 1000 s across all conditions. For the no prompt engineering condition, there is greater dispersion of points, followed by the few-shot and hybrid conditions, while the VQL condition shows greater clustering. The largest outliers were evaluated in the no prompt engineering condition.

In Fig. 2(C), the completion time by condition for the visualization technique, each point corresponds to an individual participant's execution time under a specific prompting technique. For no prompt engineering, the largest distribution was shown for bars chart and scatterplot, the line chart shows a more concentrated distribution.

In the few-shot condition, the largest distribution was for the scatterplot technique, followed by the bar chart. The Line showed the smallest distribution. For the VQL condition, the largest distribution was for the bar chart. The line chart and scatterplot showed the smallest distribution. In the hybrid Condition, the largest distribution was for the scatterplot technique, followed by the line chart, and the scatterplot showed the smallest distribution.

For the distribution of task types comparison and identification, Fig. 2(D), the values are concentrated below 1000 s across all conditions, although outliers extend this range. For no prompt engineering, there is the largest dispersion for the identification task. The few-shot condition showed slight dispersion in task

comparison. The VQL condition showed a wider spread in the Identification task, and the hybrid Condition showed the largest distribution in the comparison task.

Based on the results, it is possible to understand that the prompt engineering strategy does not affect the mean instruction construction time, although small variations may occur across conditions. Regarding the dispersion of participant data, no prompt engineering is the least consistent (highest dispersion), while VQL is the most stable.

Considering visualization techniques, the scatterplot is the most unstable, and the line chart is the most consistent, regardless of the engineering technique. It is important to note that the VQL technique reduced variability but did not eliminate the effect of the visualization technique. However, the VQL technique showed improved performance stability, especially in the comparison task.

5.2 RQ2: Prompt Structure and Complexity

The analysis of prompt structure was conducted using the Prompt Complexity variable, through the Directed Content Analysis methodology, based on the codebook presented in Table 1, Sect. 4.5. Figure 3 shows the distribution of the analysis of prompt complexity, score, across different conditions: No Prompting Engineering, Few-shot, VQL, and Hybrid, visualization types, and tasks (Comparison (C) and Identification (I)). Figure 3 (A) Performance comparing prompt strategies, Fig. 3 (B) Performance by task, for bar chart, Fig. 3 (C) Performance by task, for line chart, Fig. 3 (D) Performance by task, for scatterplot.

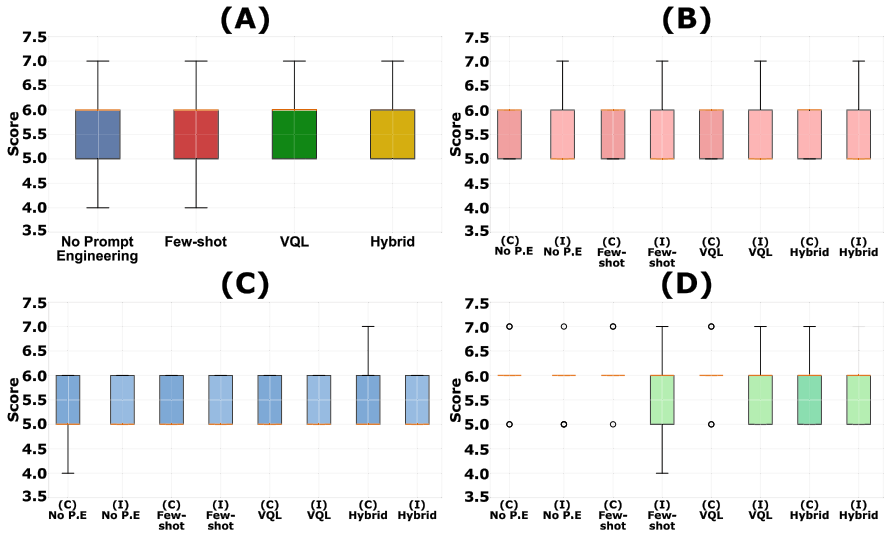

Fig. 3. Performance comparison between prompt strategies

The results for prompt complexity, showed in Fig. 3 (A), exhibited limited variation across conditions, with mean values ranging from 5.5 to 6.0. This pat-

tern suggests that, for all conditions, participants produced descriptions with similar levels of structure. The 1st quartile was flattened in the VQL and hybrid conditions, indicating that at least 50% of the data lie at or above the median.

In the bar chart, stimulus complexity, Fig. 3 (B), showed stable performance scores between the comparison (C) and identification (I) tasks, for the four conditions analyzed. However, the comparison (C) task was more stable than the identification (I) task in all conditions. This condition is characterized by the absence of the 1st and 3rd quartiles in the comparison (C) tasks in all cases, whereas, for the identification task, only the 1st quartile was flattened, remaining very close to the median.

In the line chart, Fig. 3 (C), prompt complexity showed a stable performance between tasks comparison (C) and identification (I). For the task comparison in conditions: No prompting engineering and hybrid, the 3rd and 1st quartiles were flattened, respectively.

For the scatter plot, prompt complexity, Fig. 3 (D), the tasks: comparison (C) and identification (I), for the conditions: No prompt engineering, few-shot and VQL are the same value (6.0), some outliers showed scores of 5.0 or 7.0, for the prompt complexity (score). The Identification (I) task in the few-shot condition was the most unstable condition, with scores ranging from 4.0 to 7.0. Furthermore, the Identification Task for the VQL and hybrid condition, and the Comparison for the hybrid Condition, showed the absence of the 1st quartile.

In general, the complexity of the prompts remains stable across all conditions, with limited variations related to visualization technique and task type. The greatest variability was observed in the scatterplot technique and was influenced by comparison (C) and identification (I) tasks associated with no prompt Engineering condition, suggesting that complexity is also influenced by cognitive factors in prompt construction.

5.3 RQ3: Performance Variability and Consistency

Performance variability was analyzed using dispersion measures, including standard deviation (SD), interquartile range (IQR), and maximum value (Max), considering the Completion Time variable. The Table 2 presents the main descriptive measures of variability by experimental condition.

Table 2. Variability measures of Completion Time by experimental condition.

Condition	Mean Time(s)	Standard Deviation	IQR	Max
No Prompt Engineering	370.95	426.26	338.75	2304
Few-shot	309.67	370.76	308.00	2116
VQL	307.27	298.51	241.50	1716
Hybrid	325.78	317.17	305.50	1716

The results reveal relevant differences in data dispersion across conditions. The no prompt engineering condition exhibited the highest variability, with a high standard deviation (426.26), the largest interquartile range (IQR = 338.75), and the highest observed maximum value (2304 s), indicating a strong influence of outliers and high inconsistency across participants.

The few-shot prompting showed increased variability (SD = 370.76; IQR = 308.00) in relation to VQL, demonstrating more consistent performance. Hybrid engineering showed moderate variability, thus achieving intermediate behavior.

This suggests that even if prompt engineering does not have a strong impact on completion time (RQ1), it does influence performance consistency. In this sense, structured engineering, in this case VQL and hybrid approaches, can promote more consistent results among participants and reduce potential uncertainties.

This analysis complements the results from RQ1 and RQ2, suggesting diversity in the prompt construction process, in terms of prompt complexity and prompt time, there were no significant differences between the methodologies. As shown in Fig. 2 (Sect. 5.1), there is substantial dispersion in execution times across all experimental conditions, with the presence of outliers and wide variation among participants.

5.4 RQ4: Influence of Visualization Technique and Analytical Task

The cross-analysis between visualization techniques (bar charts, line charts, and scatterplots) and analytical tasks (comparison and identification) revealed distinct effects on participant performance and prompt structure.

Regarding visualization techniques, we observed significant variation in completion time across different chart types. As shown in Table 3, bar charts exhibited the highest mean completion time (440.61 s) and the greatest dispersion (SD = 456.44), being significantly slower than line charts and scatterplots.

Table 3. Mean prompt completion time by visualization technique

Technique	Mean Time(s)	Standard Deviation
Bar Charts	440.61	456.44
Line Charts	243.72	243.41
Scatterplots	300.92	306.75

These results suggest that bar charts require greater effort during prompt construction, possibly because they require explicit specification of multiple categories and relationships between discrete variables.

To better investigate this interpretation, a multidimensional analysis was conducted. Figure 4 shows a heatmap of normalized metrics by experimental condition and variables: mean completion time (a), coefficient of variation of

completion time (b), mean prompt length (c), coefficient of variation of prompt length (d), axis mapping (e), aggregation (f), ambiguity (g), and redundancy (h).

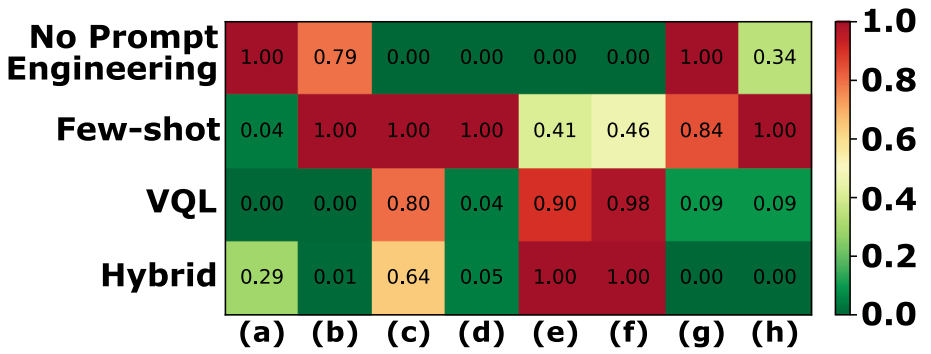

Fig. 4. Synthesis of dimensions by experimental condition.

The analysis used a normalized quantitative metrics derived from three main groups: (i) experimental performance metrics, such as mean completion time and coefficient of variation of completion time (RQ1); (ii) structural prompt metrics, such as mean prompt length and coefficient of variation of prompt length (RQ2); and (iii) qualitative metrics related to prompt content, including axis mapping, aggregation, ambiguity, and redundancy.

The results indicate that structured approaches, such as VQL and hybrid, present more balanced profiles, with higher levels of axis mapping and aggregation combined with lower levels of ambiguity and redundancy. In contrast, the no prompt engineering condition shows higher execution time, greater ambiguity, and lower structural quality of prompts, whereas the few-shot approach exhibits high variability, indicating lower consistency.

This pattern highlights that structured methodologies concentrate higher values in quality-related metrics (axis mapping and aggregation) and lower levels of ambiguity and redundancy. The hybrid methodology shows the best overall performance, while VQL combines strong structural quality with greater temporal efficiency. In contrast, the no prompt engineering condition exhibits high ambiguity and low structural performance, whereas the few-shot approach shows higher redundancy and variability.

Regarding analytical tasks, the Mann–Whitney test [10] did not indicate a statistically significant difference between comparison and identification tasks ($U = 11559.5$, $p = 0.445$). This result suggests that there is insufficient evidence to conclude that task type influences participant performance. In contrast, the observed effects across visualization techniques indicate that these constitute a more decisive factor in prompt construction effort.

The integrated analysis of the results from research questions RQ1, RQ2, and RQ4 indicated that, although the visualization technique influences the effort

required to construct the prompts, the final performance is mainly determined by the level of prompt structuring. In particular, structured engineering reduces ambiguity, increases consistency, and mitigates the negative effects associated with techniques such as bar charts.

When comparing prompt engineering methods, we observed that hybrid engineering and VQL performed best (scores of 0.858 and 0.739, respectively), combined with high axis mapping rates, greater use of aggregation, and lower redundancy, demonstrating the stability that a structured engineering approach generates. In contrast, the zero-shot (0.331) and few-shot (0.213) methods performed worst, with greater variability and lower structural quality.

Therefore, we observed that the performance of the prompts was related to the complexity of the visualization technique used and the level of structuring of the prompts.

5.5 Design Approach for Prompt Engineering in LLM-Based Data Visualization

Based on the findings of this study, we derived a set of practical approaches to support the design and use of prompt engineering strategies for LLMs performing data visualization tasks.

Structured prompt-based engineering tools like VQL should be prioritized when the output demands precision and strict control over visualization specifications. These specifications include the need for multiple variables, aggregation within the visualization, and other factors. Furthermore, we believe VQL is best suited for expert users who require reproducibility and consistency.

The few-shot approach, on the other hand, is more suitable for beginner users because it provides visual examples, helping to establish patterns in expected output formats and reducing a possible initial cognitive barrier. However, since this engineering is based only on implicit guidance, it cannot be used alone. Thus, it must be used in a hybrid manner for tasks requiring high rigor.

Structured prompts are a type of engineering that places the entire responsibility for structuring the task on the user, increasing cognitive load and the likelihood of incomplete or ambiguous prompts. Therefore, it should be applied in scenarios that involve datasets with different structures, multiple variables, or specific analytical requirements.

The results show that using unstructured prompts produces the greatest ambiguity, the highest variability, and the lowest overall performance. While acceptable for exploratory tasks, it is not suitable for the reliable generation of visualizations.

Hybrid engineering, combining the VQL structure and visual examples from a few-shot examples, provided the necessary balance to assist participants, enabling them to achieve the best overall performance in the study, as observed. Therefore, this engineering approach is suitable for user groups with varying levels of experience.

These results indicate that there is no prompt engineering strategy that is most useful. The effectiveness depends on the alignment across task complex-

ity, user experience, and the level of structural support provided. Among the approaches evaluated, the hybrid strategy emerges as a robust solution, combining structured specifications with example-based guidance to mitigate limitations.

6 Final Remarks

The results indicate that prompt engineering does not affect average user performance but does influence performance variability. Engineering strategies, such as VQL and hybrid conditions, were more effective. This result may be related to the specification of structural elements, such as axis mapping and data aggregation.

Furthermore, we found that bar charts and scatterplots require more structural specifications than line charts, influencing the level of precision needed in the instructions.

Although the quality of the resulting visualization techniques was not evaluated, the findings suggest that prompt engineering with clearer structures and less ambiguity could yield more consistent results. This interpretation should be considered with caution. These findings reinforce the importance of structuring content visualization and task categories in the prompts.

7 Limitations and Future Work

This study presents some limitations that should be considered when interpreting the results. The number of participants selected was consistent with some studies analyzed. Furthermore, the sample consisted only of students, which does not reflect the behavior of experts in the field.

Another limitation lies in the number of tasks and types of visualization. We conducted the experiment using a single language model, which may have generated the observed patterns. Different language models may exhibit distinct behaviors in the interpretation of instructions, in the structuring of responses, and in sensitivity to linguistic variations.

This study offers opportunities for future research, including qualitative analyses of the prompts constructed by participants, expanding the sample, diversifying tasks and visualization techniques, and including participants with different experience levels.

Furthermore, comparative studies across different LLMs to understand how prompt engineering approach can be applied to achieve better outputs. This support can reduce ambiguity and improve the quality of user-created instructions.

Acknowledgments. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES), Finance Code 001. The authors also acknowledge the support provided by the Dean of Research and Graduate Studies (PROPESP) of the Federal University of Pará (UFPA) through the PIAPA 2026 grant.

References

1. Aguiar, C.: Prompt-engineering-vis-data: Data for prompt engineering in AI-based data visualization (2025). https://github.com/AguiarCamile/prompt-engineering-vis-data. Accessed 14 Mar 2026
2. BNDES: BNDES data - dados e séries temporais (2025). https://www.bndes.gov.br/wps/portal/site/home/bndes-data/bndes-data-dados-series-temporais/. Accessed 14 Mar 2025
3. Chen, N., Zhang, Y., Xu, J., Ren, K., Yang, Y.: VisEval: a benchmark for data visualization in the era of large language models (2025). https://doi.org/10.1109/TVCG.2024.3456320
4. Cohen, J.: Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. Psychol. Bull. **70**(4), 213 (1968). https://doi.org/10.1037/h0026256
5. Ford, J., Zhao, X., Schumacher, D., Rios, A.: Charting the future: using chart question-answering for scalable evaluation of LLM-driven data visualizations. In: Proceedings of the 31st International Conference on Computational Linguistics, pp. 7497–7510 (2025). https://aclanthology.org/2025.coling-main.501/
6. Hsieh, H.F., Shannon, S.E.: Three approaches to qualitative content analysis. Qual. Health Res. **15**(9), 1277–1288 (2005). https://doi.org/10.1177/1049732305276687
7. Kim, N.W., Ahn, Y., Myers, G., Bach, B.: How good is ChatGPT in giving advice on your visualization design? ACM Trans. Compu.-Hum. Interact. **32**(5), 1–33 (2025). https://doi.org/10.1145/3745768
8. Li, G., et al.: Visualization generation with large language models: an evaluation. arXiv preprint arXiv:2401.11255 (2024), https://arxiv.org/abs/2401.11255
9. Ma, X., Wan, G., Yu, R., Fang, G., Wang, X.: CoT-valve: length-compressible chain-of-thought tuning. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 6025–6035 (2025). https://doi.org/10.18653/v1/2025.acl-long.300
10. Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. 50–60 (1947). https://doi.org/10.1214/aoms/1177730491
11. Monteiro, W.C., Dos Santos, D.H., De Sousa, T.A.S., Queiroz, V.F., De Araújo, T.D.O., Meiguins, B.S.: Workload evaluation to create data visualization using ChatGPT. In: 2023 27th International Conference Information Visualisation (IV), pp. 136–141 (2023). https://doi.org/10.1109/IV60283.2023.00032
12. OpenAI: OpenAI tokenizer (n.d.). https://platform.openai.com/tokenizer. Accessed 14 Mar 2026
13. Qiu, T., et al.: SmartMLVs: LLM-enabled multiple linked views generation for interactive visualization. In: 2025 IEEE 18th Pacific Visualization Conference (PacificVis), pp. 58–68 (2025). https://doi.org/10.1109/PacificVis64226.2025.00012
14. Shi, D., Xu, X., Sun, F., Shi, Y., Cao, N.: Calliope: automatic visual data story generation from a spreadsheet. IEEE Trans. Vis. Comput. Graph. **27**(2), 453–463 (2020). https://doi.org/10.1109/TVCG.2020.3030403
15. Siu, A., Kim, S.-H.G., O'Modhrain, S., Follmer, S.: Supporting accessible data visualization through audio data narratives. In: Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. CHI '22. Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3491102.3517678

16. Tian, Y., et al.: ChartGPT: leveraging LLMs to generate charts from abstract natural language. IEEE Trans. Vis. Comput. Graphics **31**(3), 1731–1745 (2024). https://doi.org/10.1109/TVCG.2024.3368621
17. Uusnäkki[1], J., Ihantola, P., Mikkonen, T.: Exploring prompt engineering with large language models. In: Arai, K. (ed.) CompCom 2025. LNNS, vol. 1423, pp. 166–177. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-92602-0_10
18. Wang, C., Lee, B., Drucker, S.M., Marshall, D., Gao, J.: Data formulator 2: iterative creation of data visualizations, with AI transforming data along the way. In: Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems. CHI '25. Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3706598.3713296
19. Wang, L., Zhang, S., Wang, Y., Lim, E.P., Wang, Y.: LLM4Vis: explainable visualization recommendation using ChatGPT. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pp. 675–692 (2023). https://doi.org/10.18653/v1/2023.emnlp-industry.64
20. Wang, Q., Zhu-Tian, C., Wang, Y., Qu, H.: A survey on ML4VIS: applying machine learning advances to data visualization. IEEE Trans. Vis. Comput. Graph. **28**(12), 5134–5153 (2022). https://doi.org/10.1109/tvcg.2021.3106142
21. Wu, Y., et al.: Automated data visualization from natural language via large language models: an exploratory study. Proc. ACM Manag. Data **2**(3), 1–28 (2024). https://doi.org/10.1145/3654992
22. Yu, H.: Leveraging prompt engineering to facilitate AI-driven chart generation. In: Proceedings of the 2024 6th International Conference on Computer Information Science and Artificial Intelligence. CISAI '24, pp. 261–268. Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3703187.3703231
23. Zhang, W., et al.: Natural language interfaces for tabular data querying and visualization: a survey. IEEE Trans. Knowl. Data Eng. **36**(11), 6699–6718 (2024). https://doi.org/10.1109/TKDE.2024.3400824
24. Zhao, Y., et al.: LightVA: lightweight visual analytics with LLM agent-based task planning and execution. IEEE Trans. Vis. Comput. Graph. **31**(9), 6162–6177 (2024). https://doi.org/10.1109/TVCG.2024.3496112