When Users Don't Specify: How LLMs Make Design Decisions in Underspecified Data Visualization Contexts

Jaeseong Ju*† Jaeeun Seo*† Jiwon Jang*† Dokyung Lee*† Hyunwoo Park‡

Seoul National University

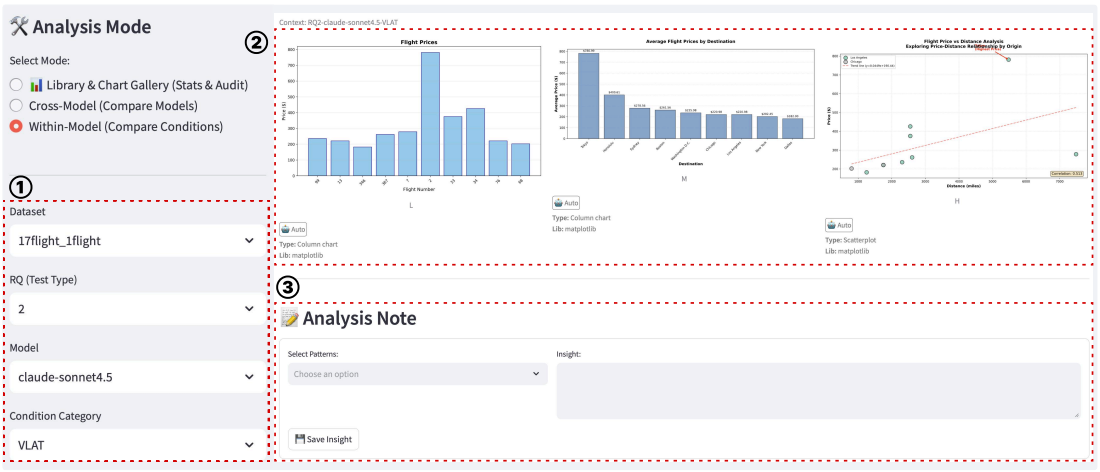

Figure 1: The interactive analysis interface for qualitative inspection. (1) A control panel for selecting dataset, model, and experimental conditions; (2) a gallery view displaying outputs across condition levels (here, visualization literacy Low to High); and (3) an annotation panel for recording observations.

Abstract

Leveraging natural language understanding and code generation capabilities, LLMs have been actively adopted for data visualization research. However, existing research primarily focuses on code correctness and execution success under well-specified prompts. In practice, users often request charts without specifying types, styles, or analytical goals. This study investigates how LLMs behave in such underspecified contexts. We audited 2,160 visualizations generated by GPT-5.2, Gemini 3 Flash, and Claude Sonnet 4.5 across varying expertise, literacy, and language conditions. Our findings reveal that under default prompting conditions, LLMs exhibit strong preferences for specific library and chart types. They tend to equate expertise levels with specific library choices and frequently add trendlines for high literacy users. Language variations showed negligible structural impact but exposed technical gaps in non-English font handling. These findings demonstrate that technical robustness does not ensure behavioral neutrality. We advocate for developing tools that steer LLM defaults toward intent-aware visualization design.

Index Terms: Natural language visualization, Large language models, Visualization design, Behavioral analysis.

1 Introduction

The rapid evolution of Large Language Models (LLMs) has accelerated innovation across diverse disciplines. Consequently, the application of LLMs to data visualization has emerged as a natural progression [13, 4, 31]. Existing research has made significant strides in lowering barriers to visualization by leveraging LLMs to offload design knowledge [28]. However, prior studies have predominantly focused on quantitative performance metrics [17, 29]. They typically optimize for accuracy against ground truth or execution success rates using goal-oriented prompts. While essential for benchmarking, this approach may not capture the complexity of real-world interactions. End-users frequently initiate requests with underspecified prompts lacking clear constraints. Our research diverges from this performance-centric view. We explore realistic ambiguity to investigate how LLMs behave when the "correct answer" is not predefined.

Understanding LLM behaviors under underspecified conditions is important from two perspectives. From a visualization perspective, vague requests effectively delegate complex design decisions to the AI. It is imperative to understand whether autonomous choices align with effective principles or revert to generic defaults. From an AI alignment perspective, these scenarios reveal latent tendencies such as library preferences or over-reliance on simple chart types. To systematically audit this autonomous decision-making, we formulate three research questions. First, how do LLMs make design decisions when prompted with underspecified requests? Second, how do contextual factors, including expertise role-playing, visualization literacy, and language context, influence generated visualizations? Third, what implicit biases or default tendencies exist across different LLM families?

To address these questions, we adopted and extended an experimental framework focused on open-ended generation. Our methodology shifts focus to underspecified prompts by deliberately omitting constraints. We conducted an API-based evaluation across three LLMs: GPT-5.2, Gemini 3 Flash, and Claude Sonnet 4.5 [16, 5, 1]. Using 60 CSV datasets, we obtained a total of 2,160

*Equal contribution
†e-mail: {cg3731, jaeeunseo, jjw6424, didrodldk}@snu.ac.kr
‡e-mail: hyunwoopark@snu.ac.kr, corresponding author

chart generations. Unlike rigid classification tasks, we treated generation as a design process. We analyzed behaviors under a baseline underspecified condition and compared them across variations in expertise, visualization literacy, and language contexts.

Our analysis reveals distinct behavioral patterns. Under baseline conditions, models converged heavily on fundamental types like line and bar charts, with matplotlib as the dominant library. Expertise role-playing triggered a clear split. Higher expertise produced increased information density through annotations. Notably, Gemini exhibited aggressive tool switching to seaborn. The visualization literacy condition revealed an elevated baseline effect. High Literacy prompts drove trendline additions. In some cases, Claude overrode single-chart constraints to generate multiple subplots. We also identified persistent gender-color biases across conditions. Models frequently mapped male categories to blue and female to pink hues when visualizing grouped data.

By shifting the evaluation focus from capability to behavior, this study offers three contributions. First, we provide a systematic empirical analysis of LLMs' visualization defaults. We identify how implicit biases emerge when human guidance is minimal. Second, we examine how contextual factors alter the structural and aesthetic quality of generated charts. This reveals model-specific adaptation strategies. Third, we discuss implications for visualization alignment. We advocate for context-aware tools that anticipate and steer LLM-generated visualizations toward neutral and intent-driven designs.

2 Background and Related Work

LLM-Driven Data Visualization. Research on Natural Language to Visualization (NL2VIS) has significantly advanced by leveraging the code generation capabilities of LLMs [28]. Initial frameworks like Chat2VIS [13] and LIDA [4] established modular pipelines that utilize prompt chaining for sequential generation. Subsequent approaches integrated feedback loops, user interactions, and external algorithms to enhance precision [21, 29]. More recently, the field has progressed toward autonomous agent-based systems capable of complex reasoning and debugging [31, 17]. Beyond end-to-end generation, LLMs are also applied to specialized tasks such as chart recommendation, alt text generation [9, 20], and data visualization evaluation [11, 7]. Additionally, Multi-modal LLMs are being explored for their ability to directly perceive and interpret chart images, moving beyond text-only processing [30].

However, these studies predominantly focus on performance optimization, measuring generation accuracy against specific benchmarks or ground truths. While some works have assessed intrinsic visualization literacy (e.g., VLAT) [6], they typically operate within well-defined constraints. Consequently, there remains a gap in examining LLM behaviors in underspecified contexts, where constraints are intentionally minimal. Since real-world users often provide vague prompts, understanding how models fill these gaps is critical. Unlike previous research prioritizing error-free rendering, we investigate the implicit defaults and biases LLMs exhibit when the correct answer is undefined, shifting the analytical lens from capability to tendency.

LLM evaluation for bias. A critical body of research focuses on AI alignment and safety, aiming to mitigate risks such as social bias and toxicity in LLMs [10]. To rigorously evaluate these risks, researchers have developed metrics to quantify prompt sensitivity [3] and analyzed worst-case performance to identify robustness failures under adversarial conditions [2]. In particular, specific prompting strategies have come under scrutiny. While role-playing can steer model behavior [26], it acts as a double-edged sword. Studies report that persona injection or even reasoning triggers like Chain-of-Thought [27] can inadvertently amplify latent biases or toxicity rather than reducing them [19]. These findings underscore that LLMs are not neutral engines. Instead, they are highly susceptible to the structural and semantic nuances of prompts. This necessitates rigorous behavioral auditing.

In the visualization domain, research is expanding beyond code generation to examine model behaviors. Recent studies have investigated specific visual preferences (e.g., encodings) [24] or benchmarked LLMs as evaluators to test alignment with human standards [25]. However, these investigations typically isolate specific design components or focus on rigid evaluation tasks. They rarely examine the holistic generation process in ambiguous scenarios. Consequently, there remains limited understanding of how LLMs orchestrate multiple design choices, ranging from chart type to aesthetics, when constraints are minimal. Addressing this gap, our study adopts an exploratory approach to observe how LLMs behave in underspecified contexts. By qualitatively analyzing outputs across diverse models and prompt variations, we aim to uncover the implicit defaults and potential biases that emerge when models are given the freedom to choose.

3 Method

3.1 Dataset Construction

We constructed datasets to examine how LLMs choose visualization designs without explicit instructions when presented with datasets of varying characteristics. Unlike prior work that prescribes chart types and evaluates compliance [23], we adopt an observational approach. We provide datasets without visualization-specific instructions and analyze the resulting design choices.

Dataset Design Philosophy. We do not define ground-truth chart types or judge correctness. Instead, we treat visualization as an open-ended design decision where multiple valid representations may exist depending on data properties and analytical intent [14, 12]. Our goal is to characterize the patterns and diversity of LLM-generated visualizations across different data characteristics. We use Vázquez's taxonomy of 24 chart types [23] as a descriptive framework to categorize LLM outputs. Four geospatial chart types requiring non-CSV formats were excluded, leaving 20 chart types as a shared vocabulary for analysis rather than prescriptive targets.

Dataset Sources and Characteristics. We compiled 60 CSV datasets combining real public data and LLM-generated data to reflect realistic usage scenarios. The collection integrates datasets and variants from prior LLM-to-visualization resources, public repositories, as well as LLM-generated datasets based on chart examples, to span heterogeneous domain contexts and encourage diverse chart choices. To cover 20 chart-type families, we curated three datasets per family ($20 \times 3 = 60$), varying key characteristics such as the number of variables (2–6), data types, and dataset size (5–500 rows). We also included a small set of minimal synthetic datasets (5–10 rows, 2–3 columns) to isolate basic visualization behavior. Detailed dataset metadata is provided in the supplemental material and our public release.

3.2 Prompt Design

To inspect LLMs' visualization decision-making within different contextual settings, we designed a structured prompt that provides visualization context through role-playing and personalization strategies [22], and conditioned the language context. We defined four prompt conditions as follows:

- **Default Prompts**: Specify only the CSV dataset and a general chart generation request, deliberately excluding potential confounding factors to establish an underspecified baseline.
- **Expertise Role-Playing Prompts**: Inject data visualization expertise levels by framing the model from the creator's perspective, explicitly specifying professional competency in visualization design.

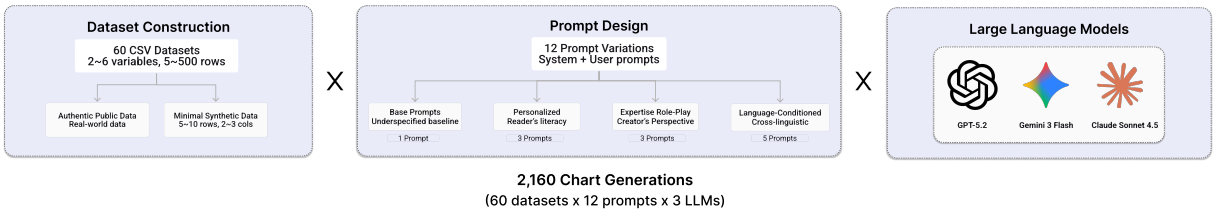

Figure 2: Systematic combination of datasets, prompt conditions, and models to produce 2,160 charts without explicit chart specifications.

- **Graph Comprehension Personalized Prompts**: Incorporate graph comprehension dimensions grounded in Curcio's visualization literacy theory [8], personalizing visualization generation on the reader's literacy level.
- **Language-Conditioned Prompts**: Specify language settings to examine cross-linguistic and cross-national variations.

To systematically control experimental variables, we designed prompt configurations as shown in Table 1. We employed a prompt structure that separates system prompts from user prompts, enabling controlled observation of the model's generation behavior [32, 15]. User prompts intentionally omit explicit chart type specifications, allowing us to capture the LLM's intrinsic tendencies in visualization decision-making [18]. Additionally, to simulate real-world scenarios involving CSV file data, we converted the metadata and content into console-friendly tabular representations to facilitate seamless processing by the LLM. We included this data representation in the prompt to approximate machine-readable format input [28], since our API-based experimental settings required text-based input rather than direct file attachments.

Table 1: Prompt configurations for chart generation

Task	Type	Prompt Content
Default	System	Give me only a python script as a response, with only a single chart.
	User	Write a python script that generates a chart using CSV file {csv_filename}. CSV file metadata: {csv_meta}. Full CSV data: {csv_full_content}.
Expertise	System	Give me only a python script as a response, with only a single chart. You are a {level} in data visualization.
	User	Write a python script that generates a chart using CSV file {csv_filename}. CSV file metadata: {csv_meta}. Full CSV data: {csv_full_content}.
Graph Comprehen.	System	Give me only a python script as a response, with only a single chart.
	User	I'm at the {curcio_level} level according to Curcio's data literacy standards. Write a python script that generates a chart using CSV file {csv_filename}. CSV file metadata: {csv_meta}. Full CSV data: {csv_full_content}
Language-Conditioned	System	Give me only a python script as a response, with only a single chart.
	User	CSV 파일 {csv_filename}을 사용하여 하나의 차트를 생성하는 파이썬 스크립트를 작성하라. CSV 파일 메타데이터: {csv_meta}. CSV 전체 데이터: {csv_full_content}.

** For language-conditioned experiments, the Default prompt was translated into Korean, Japanese, Chinese, and Spanish. This table shows only the Korean version.*

3.3 Testing Procedure

To understand recent LLMs' tendencies within data visualization, we selected three popular LLMs: GPT-5.2 (OpenAI) [16], Gemini 3 Flash (Google) [5], and Claude Sonnet 4.5 (Anthropic) [1]. We conducted experiments via standardized API parameters in order to systematically ensure consistent model versions across trials.

Our experiment yielded a comprehensive dataset of 2,160 chart generations, derived from the factorial combination of 60 datasets (20 chart-type families × 3 datasets each) × 12 prompt variations × 3 LLM models. This scale enables robust statistical analysis of visualization preferences, identifying dataset-specific and model-specific biases. Each experimental trial followed a three-step procedure:

1. **Prompting the Model**: For each experimental condition, we initialized a fresh conversation context and submitted the conditioned prompt together with the corresponding CSV dataset.
2. **Script Generation**: We collected the Python visualization scripts generated by the model, which encode visualization design decisions for the given dataset and prompt condition.
3. **Execution and Storage**: Each generated script was executed, and the resulting chart was saved as a PNG file.

3.4 Data Analysis Pipeline

Automated Metadata Extraction. We employed an LLM-as-a-Judge approach using Gemini 3 Flash to categorize the generated visualization scripts. A strict system prompt containing a taxonomy of 20 distinct chart types was used to analyze the code structure. The model identified the primary visualization library and the specific chart type.

Interactive Verification and Refinement. Following the automated labeling, we conducted a full-dataset manual audit ($N = 2,160$) via our interactive dashboard. We corrected 65 misclassifications ($\approx 3.0\%$) to ensure accuracy. All quantitative analyses in this study use these human-verified labels.

Qualitative Visual Analysis. We performed a qualitative inspection using the dashboard's Gallery Mode, which supports two complementary comparative strategies. The Cross-Model view compares outputs from different models under identical conditions to surface model-specific behaviors. The Within-Model view fixes a single model to examine variations across research factors.

To capture nuances beyond aggregate counts, four researchers independently annotated each output within the interface using lightweight tags (e.g., salient differences, consistency/inconsistency across conditions, and format compliance) alongside brief free-text notes. We then consolidated annotations through iterative group discussions to reach consensus on the key findings. The analysis dashboard, annotation logs, and the full dataset are publicly available (see Supplemental Materials).

4 Findings

Findings are organized by prompt factor. Key frequencies are reported inline, and consolidated summaries are provided in the Supplemental Materials.

4.1 Baseline Behaviors: Convergence in Types and Implementation

Our dataset was designed to accommodate 20 distinct chart types without enforcing a single golden rule, allowing for multiple valid visualization strategies per task. However, the generated outputs showed a marked convergence. In the default setting (N=180), models concentrated on a small set of fundamental formats (column: 36; line: 34; grouped column: 25; scatter: 22), even when the data structure allowed for alternative representations. Notably, within bar-type visualizations, models demonstrated a structural preference for vertical orientations: column-based charts (36+25=61) were selected far more often than horizontal bar charts (14) and grouped bars (6), suggesting that vertical layouts are treated as the primary default in unspecified contexts.

Implementation choices reflected a similar standardization. Matplotlib served as the dominant engine (144/180) and acted as the primary default across most models. In contrast, seaborn was selectively applied to specialized non-bar chart types, indicating a latent capability for tool switching based on chart complexity.

Aesthetic decisions also followed consistent patterns. In qualitative inspection, single-variable plots often retained default blue hues (e.g., #1f77b4), and multi-category plots typically relied on automatic color cycles. A notable exception emerged for gender-related data: models sometimes deviated from default cycles to apply stereotypic mappings (blue for male and pink for female) even without explicit prompting. This bias was observed in 2/4 gender-grouped bar charts (both produced by GPT-5.2), 5/5 gender-grouped column charts (including GPT-5.2's output), and 3/3 pyramid charts.

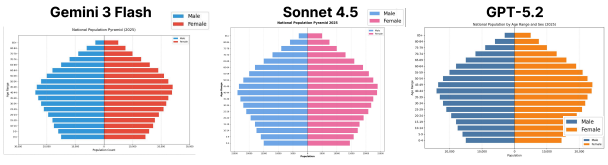

Figure 3: Implicit gender-color bias in default generations. Comparison of population pyramids from three models, using the same dataset without color specifications.

4.2 Stereotyping Expertise: Structural Shifts and Tool Switching

Injecting expertise prompts triggered a distinct split, with Medium and High (M–H) behaving similarly and diverging from Low (L). In L, vertical column charts dominated (60/180; 33%), exceeding even the baseline share. In contrast, M–H conditions sometimes increased information density, e.g., evolving from simple scatter plots to bubble charts to encode additional variables. Bar charts in M–H also almost always included explicit numeric value annotations (M: 18/18; H: 23/25 rendered). Gender-related charts exhibited semantic color biases across expertise levels: for grouped column charts, stereotypic mappings (male→blue, female→pink/red) appeared in 5/6 cases in M and 3/6 in H, while reversals accounted for 1/6 (M) and 3/6 (H). In L, the bias was even more dominant (9/10). A similar pattern appeared in pyramid charts (L: 1/1; M: 4/4; H: 5/5).

Library selection revealed model-dependent interpretations of expertise. Gemini exhibited the most drastic strategy shift: while it relied on matplotlib in L, it switched to seaborn in M–H (>55% of outputs), effectively associating seaborn with "expert-level" visualization. In contrast, Claude maintained a strong preference for matplotlib across all levels and instead expressed expertise through styling adjustments (e.g., bold titles, refined axis labels). GPT-5.2 showed less consistent shifts in library choice and structural complexity across expertise levels.

The distinction in visualization logic was most apparent in more complex cases. In qualitative inspection, we observed that M–H prompts sometimes led models to use grouping/aggregation to produce segmented summaries and to incorporate additional encodings, occasionally alongside minimalist styling choices such as reduced gridlines (notably in Gemini). These examples suggest that "expert-like" outputs may involve both added information and refinement in presentation.

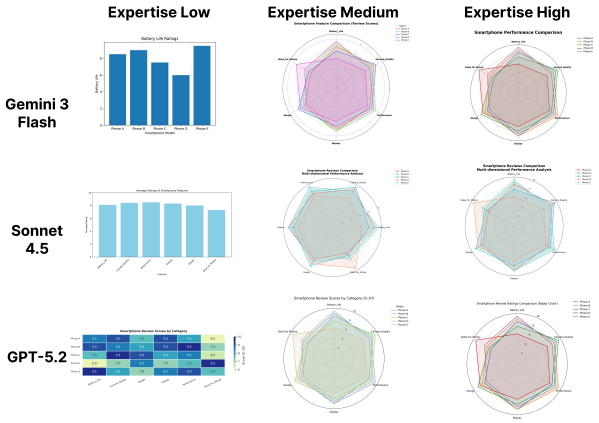

Figure 4: Effect of expertise level prompts on chart generation. Low expertise yields simpler chart types, while Medium and High levels converge toward more complex representations such as radar charts.

4.3 Adapting to Literacy: Elevated Baselines and Insight Prioritization

The visualization literacy condition shared broad similarities with the Expertise experiment, particularly in the dominance of column-based charts at the Low literacy level (N=180; column: 90; grouped column: 27) and the Gemini-driven increase in seaborn usage from Low to High. Across all models, seaborn usage rose from 6 (L) to 14 (M) to 34 (H) outputs, driven primarily by Gemini (3, 12, and 33, respectively). However, the progression between levels differed from Expertise. Unlike the clearer "Low vs. Medium–High" split observed in the Expertise prompts, literacy adaptation appeared more gradual: Low and Medium outputs were often more similar in structure across models, suggesting an "elevated baseline" for literacy. Models seemed to assume a higher minimum standard for "Low Literacy" than for "Low Expertise," resulting in less abrupt structural changes until the High level.

The transition to High literacy triggered more explicit attempts to deliver insights. Given that the High prompt framed the user as capable of "predicting future trends," models (particularly Claude and Gemini) frequently appended trendlines or statistical cues to line and scatter plots (Gemini: 31/60; Claude: 19/60, emerging predominantly at High). Gemini also occasionally emphasized specific data regions via additional visual adjustments (e.g., background shading or grid refinements). Most notably, Claude sometimes prioritized insight delivery over system constraints: despite the instruction to generate a "single chart," it produced multiple subplots in 12/60 (20%) cases at the High level, indicating that perceived user literacy can override structural constraints in the system prompt.

While Gemini and Claude showed identifiable patterns, GPT-5.2 exhibited a non-linear progression across literacy levels. Its Medium outputs occasionally contained higher information density or more complex elements than High, making a consistent adaptation strategy difficult to discern. In styling, Claude showed a lower threshold for visual refinement in the literacy condition than in Expertise, applying bold title formatting even at Low. However, per-

sistent biases remained: gender-color mappings continued to appear across literacy levels in Claude, suggesting that literacy adaptation does not necessarily correct underlying stereotypic tendencies.

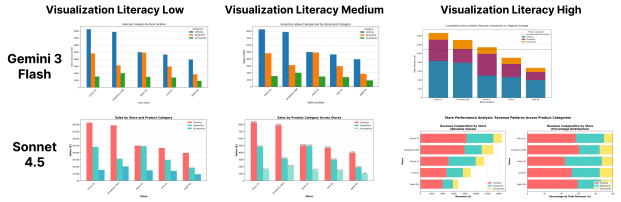

Figure 5: Effect of visualization literacy prompts on chart generation. Low and Medium levels produce similar outputs, while High triggers divergent behavior, including multi-subplot responses from Sonnet.

4.4 Linguistic Invariance and Localization Configuration Challenges

Variations in linguistic context resulted in negligible structural adaptation. This contrasts with the shifts observed in the Expertise and visualization literacy conditions. The distribution of chart types remained consistent with the baseline and maintained a strong preference for column, line, and scatter plots across all languages. Minor fluctuations occurred, such as a slight increase in grouped charts for Spanish (27 cases). However, these did not constitute a significant deviation. Instead of distinct stylistic changes, the primary challenge lay in technical configuration for non-English content. We utilized translated datasets and localized system prompts to induce linguistic adaptation. Despite this, models frequently failed to generate necessary font configuration codes, such as CJK settings in matplotlib. Consequently, while the visualization logic remained intact, the outputs often suffered from encoding errors ("tofu" glyphs). This indicates that language prompts primarily affect content translation rather than visual structure.

4.5 Code Generation Reliability and Error Taxonomy

Despite these configuration challenges, generated code was generally robust. Across 2,160 generations (3 models × 720 tasks), only 28 cases (1.3%) resulted in execution failures. Error patterns were largely model-specific rather than attributable to prompt complexity. GPT-5.2 was the most stable (4/720 failures), with rare type/attribute errors often tied to invalid inputs. Claude (11/720) mainly struggled with dependency management (e.g., using np without importing NumPy or calling japanize_matplotlib without proper imports). Gemini (13/720) failed mostly during data ingestion (e.g., CSV parsing logic or incorrect argument passing). Overall, failures reflected basic oversights or environment mismatches, suggesting that the remaining bottlenecks are less about generating runnable plotting code and more about environmental adaptability and visual refinement.

5 Discussion

Our results confirm that LLMs have matured into robust visualization generators. They exhibited high execution success rates even with underspecified prompts. However, this technical robustness does not imply neutrality. When prompts lack explicit constraints, models rely on specific defaults. These include a preference for fundamental statistical charts and particular library choices. These defaults are not random. They reflect latent assumptions about "standard" visualization. We observed persistent gender-color biases and distinct behavioral shifts when expertise or literacy personas were injected. This underscores the necessity of explicit constraint specification. We cannot assume that an "underspecified" prompt will yield a "neutral" result. Instead, it triggers the model's intrinsic biases. Therefore, developers must account for these model-specific traits. They should implement intent-driven steering strategies to control complexity and mitigate stereotypic patterns. Additionally, technical gaps in non-English contexts suggest that prompting alone is insufficient for localization. Future systems require API-level middleware to automatically handle environmental dependencies, such as font configurations, to ensure true global adaptability.

This study serves as an exploratory investigation and implies several limitations regarding scope and rigor. A primary limitation lies in the evaluation of visualization quality. We analyzed behavioral shifts, such as library switching or the inclusion of trendlines. However, we did not assess whether these outputs were empirically "better" or truly suitable for the target expertise and literacy levels. Establishing ground truth for "expert" or "high-literacy" visualizations remains a complex challenge. It requires extensive human-subject evaluation. Additionally, our experimental design relied on a fixed dataset and limited repetition. Future work should aim for greater methodological rigor by expanding the dataset size and incorporating multi-turn interactions. We also plan to refine prompt variables to verify statistical significance. Despite these limitations, this work offers a meaningful first step in uncovering how LLMs navigate ambiguity. We highlighted the implicit biases and decision-making patterns that emerge in underspecified contexts. We hope this prompts the community to be more mindful of the latent characteristics embedded in AI-assisted visualization tools.

6 Conclusion

This study explored the unexamined behaviors of LLMs in underspecified data visualization contexts. Our findings confirm that while modern LLMs have achieved near-perfect technical robustness in code execution, they demonstrate distinct latent tendencies that frame how data is represented. When human guidance is minimal, models often rely on associative heuristics. These include linking expertise with visual complexity, prioritizing insight delivery over faithfulness, and maintaining structural rigidity despite linguistic changes.

These results highlight that an "underspecified" prompt is never truly neutral. It reflects the model's training patterns, ranging from library preferences to observed gender-color associations. The challenge for future AI-assisted visualization is not just whether the model can generate a chart, but how it navigates these choices. We conclude that achieving trustworthy visualization requires moving beyond syntax checking to behavioral alignment. Future systems must incorporate environment-aware middleware and intent-driven steering mechanisms to ensure that the "Implicit Analyst" facilitates objective decision-making tailored to user needs.

References

[1] Anthropic. Claude Sonnet 4.5. https://www.anthropic.com/claude/sonnet, 2026. Accessed via Anthropic API. 1, 3

[2] B. Cao, D. Cai, Z. Zhang, Y. Zou, and W. Lam. On the worst prompt performance of large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, eds., *Advances in Neural Information Processing Systems*, vol. 37, pp. 69022–69042. Curran Associates, Inc., 2024. doi: 10.52202/079017-2205 2

[3] A. Chatterjee, H. S. V. N. S. K. Renduchintala, S. Bhatia, and T. Chakraborty. POSIX: A prompt sensitivity index for large language models. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 14550–14565. Association for Computational Linguistics, Miami, Florida, USA, Nov. 2024. doi: 10.18653/v1/2024.findings-emnlp.852 2

[4] V. Dibia. LIDA: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models. In D. Bollegala, R. Huang, and A. Ritter, eds., *Proceedings of*

the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pp. 113–126. Association for Computational Linguistics, Toronto, Canada, July 2023. doi: 10.18653/v1/2023.acl-demo.11 1, 2

[5] Google. Gemini 3 Flash Model. https://ai.google.dev/gemini-api/docs/gemini-3, 2026. Accessed via Gemini API. 1, 3

[6] J. Hong, C. Seto, A. Fan, and R. Maciejewski. Do LLMs have visualization literacy? an evaluation on modified visualizations to test generalization in data interpretation. *IEEE Transactions on Visualization and Computer Graphics*, 31(10):7004–7018, 2025. doi: 10.1109/TVCG.2025.3536358 2

[7] N. W. Kim, Y. Ahn, G. Myers, and B. Bach. How good is ChatGPT in giving advice on your visualization design? *ACM Trans. Comput.-Hum. Interact.*, 32(5), Oct. 2025. doi: 10.1145/3745768 2

[8] S. Lee, S.-H. Kim, and B. C. Kwon. VLAT: Development of a Visualization Literacy Assessment Test. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):551–560, 2017. doi: 10.1109/TVCG.2016.2598920 3

[9] X. Li, J. Zhou, W. Chen, D. Xu, T. Xu, and E. Chen. Visualization recommendation with prompt-based reprogramming of large language models. In L.-W. Ku, A. Martins, and V. Srikumar, eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13250–13262. Association for Computational Linguistics, Bangkok, Thailand, Aug. 2024. doi: 10.18653/v1/2024.acl-long.716 2

[10] Y. Liu, Y. Yao, J.-F. Ton, X. Zhang, R. Guo, H. Cheng, Y. Klochkov, M. Faaiz Taufiq, and H. Li. Trustworthy LLMs: a Survey and Guideline for Evaluating Large Language Models' Alignment. *arXiv e-prints*, p. arXiv:2308.05374, Aug. 2023. doi: 10.48550/arXiv.2308.05374 2

[11] L. Y.-H. Lo and H. Qu. How good (or bad) are LLMs at detecting misleading visualizations? *IEEE Transactions on Visualization and Computer Graphics*, 31(1):1116–1125, 2025. doi: 10.1109/TVCG.2024.3456333 2

[12] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, Apr. 1986. doi: 10.1145/22949.22950 2

[13] P. Maddigan and T. Susnjak. Chat2VIS: Generating data visualizations via natural language using ChatGPT, codex and GPT-3 large language models. *IEEE Access*, 11:45181–45193, 2023. doi: 10.1109/ACCESS.2023.3274199 1, 2

[14] T. Munzner. Visualization analysis and design. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Courses*, SIGGRAPH Courses '25. Association for Computing Machinery, New York, NY, USA, 2025. doi: 10.1145/3721241.3733989 2

[15] A. Neumann, E. Kirsten, M. B. Zafar, and J. Singh. Position is power: System prompts as a mechanism of bias in large language models (LLMs). In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '25, p. 573–598. Association for Computing Machinery, New York, NY, USA, 2025. doi: 10.1145/3715275.3732038 3

[16] OpenAI. GPT-5.2 Model. https://platform.openai.com/docs/models/gpt-5.2, 2026. Accessed via OpenAI API. 1, 3

[17] G. Ouyang, J. Chen, Z. Nie, Y. Gui, Y. Wan, H. Zhang, and D. Chen. nvAgent: Automated data visualization from natural language via collaborative agent workflow. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19534–19567. Association for Computational Linguistics, Vienna, Austria, July 2025. doi: 10.18653/v1/2025.acl-long.960 1, 2

[18] A. Sedova, R. Litschko, D. Frassinelli, B. Roth, and B. Plank. To know or not to know? analyzing self-consistency of large language models under ambiguity. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 17203–17217. Association for Computational Linguistics, Miami, Florida, USA, Nov. 2024. doi: 10.18653/v1/2024.findings-emnlp.1003 3

[19] O. Shaikh, H. Zhang, W. Held, M. Bernstein, and D. Yang. On second thought, let's not think step by step! bias and toxicity in zero-shot reasoning. In A. Rogers, J. Boyd-Graber, and N. Okazaki, eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4454–4470. Association for Computational Linguistics, Toronto, Canada, July 2023. doi: 10.18653/v1/2023.acl-long.244 2

[20] N. Singh, L. L. Wang, and J. Bragg. FigurA11y: AI assistance for writing scientific alt text. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, IUI '24, p. 886–906. Association for Computing Machinery, New York, NY, USA, 2024. doi: 10.1145/3640543.3645212 2

[21] 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*, 31(3):1731–1745, 2025. doi: 10.1109/TVCG.2024.3368621 2

[22] Y.-M. Tseng, Y.-C. Huang, T.-Y. Hsiao, W.-L. Chen, C.-W. Huang, Y. Meng, and Y.-N. Chen. Two tales of persona in LLMs: A survey of role-playing and personalization. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16612–16631. Association for Computational Linguistics, Miami, Florida, USA, Nov. 2024. doi: 10.18653/v1/2024.findings-emnlp.969 2

[23] P.-P. Vázquez. Are LLMs ready for visualization? In *2024 IEEE 17th Pacific Visualization Conference (PacificVis)*, pp. 343–352, 2024. doi: 10.1109/PacificVis60374.2024.00049 2

[24] H. W. Wang, M. Gordon, L. Battle, and J. Heer. DracoGPT: Extracting visualization design preferences from large language models. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):710–720, 2025. doi: 10.1109/TVCG.2024.3456350 2

[25] H. W. Wang, J. Hoffswell, S. M. Thazin Thane, V. S. Bursztyn, and C. X. Bearfield. How aligned are human chart takeaways and LLM predictions? a case study on bar charts with varying layouts. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):536–546, 2025. doi: 10.1109/TVCG.2024.3456378 2

[26] N. Wang, Z. Peng, H. Que, J. Liu, W. Zhou, Y. Wu, H. Guo, R. Gan, Z. Ni, J. Yang, M. Zhang, Z. Zhang, W. Ouyang, K. Xu, W. Huang, J. Fu, and J. Peng. RoleLLM: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. In L.-W. Ku, A. Martins, and V. Srikumar, eds., *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14743–14777. Association for Computational Linguistics, Bangkok, Thailand, Aug. 2024. doi: 10.18653/v1/2024.findings-acl.878 2

[27] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou. Chain-of-Thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds., *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824–24837. Curran Associates, Inc., 2022. 2

[28] Y. Wu, Y. Wan, H. Zhang, Y. Sui, W. Wei, W. Zhao, G. Xu, and H. Jin. Automated data visualization from natural language via large language models: An exploratory study. *Proc. ACM Manag. Data*, 2(3), May 2024. doi: 10.1145/3654992 1, 2, 3

[29] Y. Xie, Y. Luo, G. Li, and N. Tang. HAIChart: Human and AI paired visualization system. *Proc. VLDB Endow.*, 17(11):3178–3191, July 2024. doi: 10.14778/3681954.3681992 1, 2

[30] Z. Xu and E. Wall. Exploring the capability of LLMs in performing low-level visual analytic tasks on SVG data visualizations. In *2024 IEEE Visualization and Visual Analytics (VIS)*, pp. 126–130, 2024. doi: 10.1109/VIS55277.2024.00033 2

[31] Z. Yang, Z. Zhou, S. Wang, X. Cong, X. Han, Y. Yan, Z. Liu, Z. Tan, P. Liu, D. Yu, Z. Liu, X. Shi, and M. Sun. MatPlotAgent: Method and evaluation for LLM-based agentic scientific data visualization. In L.-W. Ku, A. Martins, and V. Srikumar, eds., *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11789–11804. Association for Computational Linguistics, Bangkok, Thailand, Aug. 2024. doi: 10.18653/v1/2024.findings-acl.701 1, 2

[32] L. Zhang, T. Ergen, L. Logeswaran, M. Lee, and D. Jurgens. SPRIG: Improving Large Language Model Performance by System Prompt Optimization. *arXiv e-prints*, p. arXiv:2410.14826, Oct. 2024. doi: 10.48550/arXiv.2410.14826 3