Exploring Multimodal Prompt for Visualization Authoring with Large Language Models

Zhen Wen, Luoxuan Weng, Yinghao Tang, Runjin Zhang, Yuxin Liu, Bo Pan, Minfeng Zhu, and Wei Chen

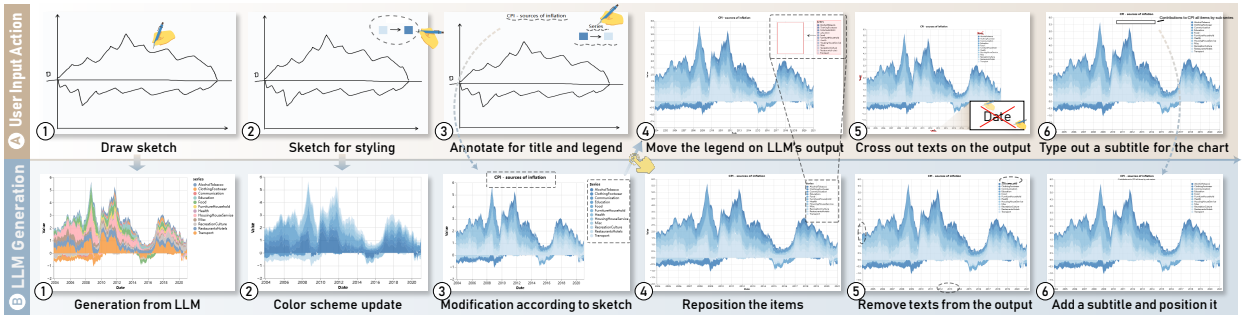

Fig. 1: Multimodal prompt for visualization authoring with VisPilot. (A) The user can create visualizations by providing sketching, text annotations or directly manipulating existing visualizations. (B) VisPilot interprets the multimodal input and generates visualizations.

Abstract—Recent advances in large language models (LLMs) have shown great potential in automating the process of visualization authoring through simple natural language utterances. However, instructing LLMs using natural language is limited in precision and expressiveness for conveying visualization intent, leading to misinterpretation and time-consuming iterations. To address these limitations, we conduct an empirical study to understand how LLMs interpret ambiguous or incomplete text prompts in the context of visualization authoring, and the conditions making LLMs misinterpret user intent. Informed by the findings, we propose a novel prompting framework that introduces visual prompts as a complementary input modality to text prompts. Our approach systematically translates visual inputs into structured, step-by-step instructions for LLMs, which help clarify user intent and improve LLMs' interpretation abilities. To validate this multimodal prompting framework, we develop VisPilot, a proof-of-concept system that embodies our approach and enables users to create visualizations through a seamless combination of multimodal inputs, including text, sketches, and direct manipulations. Through two case studies and a user study, we demonstrate that VisPilot provides a more intuitive way to create visualizations without affecting the efficiency compared to text-only prompting approaches. Furthermore, we analyze the impact of text and visual prompts in different tasks. Our findings highlight the importance of multimodal prompting in improving the usability of LLMs for visualization authoring. We discuss design implications for future visualization systems and provide insights into how multimodal prompts can enhance human-AI collaboration in creative visualization tasks. All materials are available at https://OSF.IO/2QRAK.

Index Terms—Visualization authoring, large language model, multimodal prompting

1 Introduction

Visualization authoring tools have evolved rapidly to lower the barriers of creating data visualizations, from expertise-driven languages and formal grammars [1, 21, 27] to more accessible graphical interfaces [9,23,34,37,40,41]. With the emergence of large language models (LLMs), visualization creation has been further simplified through natural language interfaces that automatically translate user utterances into visualization specifications [4, 10, 33, 46]. However, despite their accessibility, recent studies indicate that LLMs are limited in understanding accurate visualization intent from natural language inputs [2, 28].

Conveying visualization intent in natural languages faces challenges in terms of accuracy and expressiveness [20]. First, natural language inherently contains *ambiguity and implicit cues* [5, 19, 31], requiring LLMs to infer the user's true intent. Since such inferences can be ambiguous, the visualizations often deviate from the expected outcome. Second, there exists a fundamental *modality gap* between textual descriptions and graphical visualizations: users struggle to precisely articulate visual intent through text alone. Once the model generates an unexpected visualization, it is difficult for users to diagnose or correct the underlying issues through text-only instructions. These limitations highlight the need for a more effective approach to instruct LLMs with accurate visual intent in the process of visualization authoring.

To enable LLMs to more accurately understand visual intent, recent research has explored multimodal prompting in the tasks of image generation [24] and visual question answering [35]. Multimodal prompting combines textual and visual inputs to enhance the understanding of user intent, allowing for more accurate and expressive outputs. Nevertheless, the use of multimodal prompts for visualization authoring remains unexplored. It is unclear that whether visual prompts can effectively address the limitations of text prompts and how to instruct LLMs with multimodal prompts for visualization authoring tasks.

To inform our study, we conduct an empirical study to understand the limitations of using text prompts to instruct LLMs for visualization creation. Through a systematic analysis of 814 natural language utterances used to request visualizations, we find that prompting LLMs with natural language frequently leads to misinterpretation of user intent due to three reasons: (1) limited expression of visual intent: text prompts are inherently inflexible to express visual intent; (2) inadequate guidance for LLM behavior: non-expert or inaccurate expressions constantly lead to LLMs' incorrect inferences; (3) misaligned human-LLM

- *Zhen Wen, Luoxuan Weng, Yinghao Tang, Runjin Zhang, Yuxin Liu, Bo Pan, and Wei Chen are with the State Key Lab of CAD&CG, Zhejiang University. E-mail: {wenzhen, lukeweng, yinghaotang, 3210105680, 3220104160, bopan, chenvis}@zju.edu.cn.*
- *Minfeng Zhu is with Zhejiang University. Email: {minfeng_zhu}@zju.edu.cn.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

design preferences: prompts without explicit design instructions result in biased design preferences of different LLMs. To address these issues, we propose a multimodal prompting framework that incorporates visual prompts as a complement to text prompts.

We develop VisPilot as a proof-of-concept system to evaluate the feasibility of the multimodal prompting framework. The system allows users to create visual prompts by sketching on a canvas. VisPilot supports four fundamental visual input actions: scratch (freehand drawing of visualization elements), style (applying visual properties), annotation (providing textual context), and manipulation (modifying existing elements). This suite of interactions enables users to express visualization intent through visual manners, complementing traditional text prompts. The underlying prompting framework interprets these visual inputs through a step-wise reasoning process, translating sketches into precise visualization specifications. We demonstrate the effectiveness of VisPilot through two case studies on visualization authoring and data exploration, along with a user study comparing the performance of text-only and multimodal prompting for visualization authoring tasks. The results show that VisPilot achieves higher task accuracy and more satisfying user experience without significantly affecting task efficiency compared to the text-only prompting condition. Based on the findings from evaluation studies, we identify several valuable design implications for multimodal prompting in visualization authoring scenarios.

In summary, our main contributions include: (1) an empirical study identifying the limitations of text prompts for visualization authoring, (2) a prompting framework with a prototype system[1] that supports users to create visualizations using multimodal prompts, and (3) an evaluation demonstrating the impact of text and visual prompts on specifying visualizations, along with design implications for future multimodal visualization authoring systems.

2 Related Work

2.1 LLMs for Visualization Generation

LLMs have been demonstrated as a convenient interface for natural language to visualization tasks [16,38,43]. Recent research has focused on enhancing LLMs for visualization generation, with a particular emphasis on model fine-tuning and prompting strategies. ChartLlama [10] and ChartGPT [33] fine-tune language models with visualization domain knowledge, while systems like LIDA [4] and FinFlier [13] leverage carefully designed prompting strategies for visualization generation without modifying the underlying models. Several studies have proposed evaluation criteria for LLM-generated visualizations. VisEval [3] and DracoGPT [36] establish benchmarks and metrics to assess visualization quality, appropriateness, and adherence to design principles.

While these approaches have advanced LLM-based visualization generation, they primarily formulate the visualization requirement as a natural language prompt, which may not fully capture the complexity of user intent and design principles. Our work aims to address this gap by exploring multimodal prompting strategies that incorporate both visual and textual elements to enhance the generation of visualizations.

2.2 Multimodal Prompt for Generative Models

Due to the inherent ambiguity and redundancy of text prompts, researchers have explored multimodal prompt design for LLMs [47]. For example, visual prompts like colorful boxes or circles can direct multimodal large language models (MLLMs) to specific regions of interest, thereby improving their generation quality [18,42,45]. This strategy has been widely applied to computer vision tasks such as visual question answering [35], image editing [24], and knowledge tagging [8]. Recent studies have also investigated interaction-augmented prompts to facilitate precise user intent understanding [28]. Chen *et al.* [2] propose a design space for generative visual analytics and develop a direct manipulation interface. Similarly, DirectGPT [22] characterizes four direct manipulation actions to enhance the efficiency of human-LLM communication. However, these approaches typically transform interactions back to engineered text prompts, without explicitly incorporating visual inputs. Moreover, the current literature lacks a clear understanding of multimodal prompt design considerations to improve the expressiveness and efficiency of the visualization authoring process. Our work extends prior endeavors by identifying four key design principles for visual prompting and examining how different prompt modalities influence visualization specification.

2.3 Multimodal Interactions for Visualization

Multimodal interactions have been widely studied in the context of visualization systems, enabling users to create or interact with visualizations through multiple input modalities such as direct manipulations [25], freehand sketches [29], and natural language queries [26]. Due to the accessibility and extensibility of natural language, many systems have explored combining verbal and visual modalities to enhance user experience in visualization authoring. DataTone [7] and Orko [32] pioneered approaches that integrate natural language with direct manipulation interfaces, allowing users to refine ambiguous queries through interactive widgets. Similarly, tools like Valletto [15,26] facilitate natural language interactions with visualizations through contextual dialogs and touch-based interactions. Recent systems such as WYTIWYR [44] and VisLTR [12] have further advanced the integration of multimodal inputs through cross-modal neural networks, which recommend visualizations based on user queries and multimodal context. These studies have demonstrated the potential of multimodal interactions in enhancing user experience on data visualization and exploration. Informed by them, we aim to investigate how multimodal prompting can be effectively utilized in LLMs for visualization authoring.

3 Empirical Study: Text Prompt for Visualization

To understand the limitations of text prompts for visualization authoring, we conduct a systematic analysis of 814 natural language utterances used to request visualizations. The analysis identifies the key challenges that multimodal prompting approach needs to address. In particular, our empirical study aims to address the following research questions:

- ***RQ1:*** *How do LLMs interpret natural languages for visualization specifications?* The natural language utterance for creating a visualization often expresses multifaceted preferences for visualization design, such as mark type, encoding choices, etc. Previous studies [16, 30] have shown that users often explicate only partial specifications in their utterances. These observations raise the question of how LLMs infer complete visualization specifications from ambiguous or incomplete natural language utterances.
- ***RQ2:*** *What makes LLMs generate unexpected visualization specifications?* The performance of LLMs in generating visualization specifications is often unpredictable. While recent studies [3, 43] have discussed the reasons behind unexpected results in terms of prompt design and model capabilities, the effects of ambiguity and incompleteness of user utterances are still unclear. We aim to understand how the ambiguity and incompleteness in natural language expressions contribute to unexpected visualization outputs.

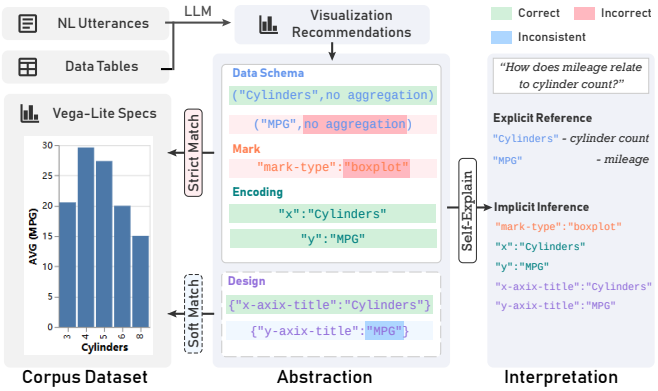

Fig. 2: The procedure of LLM processing on the corpus data.

[1]The source code and online demo: https://github.com/KidsXH/vispilot.

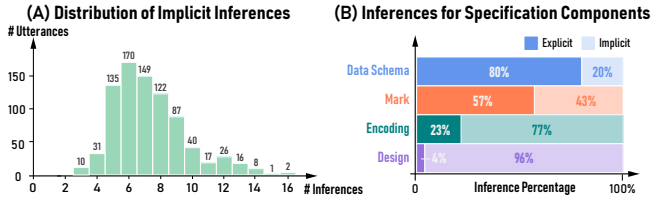

Fig. 3: The analysis results of LLM interpretation of utterances. (A) The number of implicit inferences made for each utterance, where each item of specification components is counted as a separate inference if it is inferred by the LLM. (B) The percentage of explicit references or implicit inferences made for each specification component, where a component is counted as implicit if at least one of its items is implicitly inferred.

3.1 Methodology

Data Source. We analyze the corpus of 814 natural language utterances collected by Srinivasan et al. [30]. This dataset was compiled through a structured study with 102 participants who were shown ten canonical visualizations (*e.g.*, bar charts, line charts, scatterplots) and asked to provide natural language utterances they would use to create these visualizations. Each utterance in this corpus is paired with both the data source and the target visualization written in Vega-Lite code.

Framework. To better understand how LLMs interpret natural language utterances for multifaceted visualization specifications, we establish a framework to abstract the components of specifications referring to previous work [23, 27]. This framework serves as a foundation for our analysis of the corpus and the design of our multimodal prompting approach. We define a specification S as a composition of four essential components that together determine the complete visualization design:

- **Data Schema $\mathscr{D}$.** This component defines the structural properties of data, including attribute names and data transformations. Formally, $\mathscr{D} = \{(a_i, f_i)\}$ where a_i is an attribute, and f_i is an optional aggregation function (*e.g.*, sum, average, count). The schema provides a foundation for understanding the data's structure and semantics.
- **Mark $\mathscr{M}$.** Marks constitute the fundamental graphical primitives that represent data items. Formally, $\mathscr{M} = \{m_1, m_2, ..., m_n\}$ where m_i is a mark type (*e.g.*, point, line, bar), which defines how data items are visually represented and the basic visual form of the chart.
- **Encoding $\mathscr{E}$.** This component specifies the mapping between data attributes and visual properties: $\mathscr{E} = \{(a_i, v_j)\}$ where a_i is a data attribute and v_j is a visual channel (*e.g.*, position, size, color). These mappings transform abstract data into perceptible visual forms.
- **Design $\mathscr{G}$.** The design component captures visualization properties not directly tied to data semantics, including $\mathscr{G} = \{g_1, g_2, ..., g_n\}$ where each g_i represents stylistic elements like background color, gridlines, axis properties, chart title, etc. These elements typically focus on the aesthetics and accessibility of the visualization.

Procedure. We analyze the performance of LLMs in interpreting natural language utterances to visualization specifications in two steps: (1) *LLM Processing* (Fig. 2), where we instruct the model with 814 utterances to generate visualizations through Vega-Lite specifications, followed by a self-explanation of its inference rationale and an accuracy evaluation; (2) *Expert Analysis*, where four domain experts independently analyze the results of LLM processing to identify the limitations of text prompts and the reasons behind LLM misinterpretation. Afterwards, the experts conduct a meeting to discuss and resolve any discrepancies in analysis results, addressing the two research questions.

We perform the analysis on three advanced LLMs (Gemini-2.0-Flash, GPT-4o, and Claude-3.5-Sonnet), and have consistent insights across all three models. To simplify the presentation, we mainly report the results of Gemini-2.0-Flash in this paper, while the results of the other two models are included in the supplementary material.

3.2 LLM Interpretation of Natural Language Utterances

To address ***RQ1***, we categorize the LLM interpretation patterns of natural language utterances into two types, depending on the explicitness of the specification expressed in the utterance:

- **Explicit Reference**: The LLM interprets the user intent based on sufficient rationale and explicit specification of the visualization components. For example, *"show me the average sales by region using a bar chart"* explicitly specifies the data schema $\mathscr{D} = \{(sales, average), (region, raw)\}$ and the mark type $\mathscr{M} = \{bar\}$.
- **Implicit Inference**: The LLM interprets the user intent based on implicit assumptions and incomplete specification of the visualization components. For example, *"Show me the sales by region"* does not explicitly specify the aggregation function (*e.g.*, average, sum) or the mark type (*e.g.*, bar, line), leading to implicit inference of the data schema $\mathscr{D} = \{(sales, sum), (region, raw)\}$, mark type $\mathscr{M} = \{bar\}$, and encoding $\mathscr{E} = \{(region, x), (sales, y)\}$.

We employ the self-explanation mechanism of LLMs [14] to identify these interpretation patterns. The LLM is prompted to generate a self-explanation of its reasoning process, including the rationale for its interpretation with a classification of interpretation patterns (explicit or implicit) for all components in the specification.

Results. Figure 3-A shows that the user utterances are widely ambiguous or incomplete, leading to at least three implicit inferences for each utterance ($\mu = 7.27$, $\sigma = 2.23$). The top 3 frequent inference made by the LLMs is on the mark style (814 times), x-axis encoding (724), and y-axis title (567). When the user requests the most complicated visualizations in the corpus, such as *"For each country show the relationship between average acceleration and number of cylinders"*, the LLMs need to make 16 implicit inferences to generate the target visualization, involving design (10), encoding (4), mark (1), and data schema (1) specifications. We then analyze the implicit inference patterns specifically on four specification components, as shown in Fig. 3-B.

- **Data Schema (20%).** The data schema is the most frequently explicitly specified component, with only 20% of utterances exhibiting ambiguity or incompleteness. The most frequently inferred data schema is the y-axis aggregation (151), which is often assumed to be *sum* or *average* when not explicitly stated. Besides, the users also frequently omit the third data attribute (139) in their utterances, which is often used for *color* or *column* in the encoding specification.
- **Mark (43%).** More than half of utterances manage to explicitly specify the mark type through describing the chart type (*e.g.*, bar chart, line chart) or the shape of the marks (*e.g.*, point, line), while the other 36% of utterances lead to implicit inferences on the mark type. These utterances are often expressed in vague terms like *"relationship"* or *"compare"* for their analytic tasks, which do not explicitly indicate the mark type. The most frequently inferred mark type is bar (47% of 420 bar charts), followed by point/circle (44% of 262 scatter plots) and line (38% of 95 line charts).
- **Encoding (77%).** Most utterances lead to implicit inferences on encoding choices. The users usually do not explicitly specify the encoding choices for coordinates (727 of x-axis and 509 of y-axis), which are often inferred by *"by convention"* or *"by experience"* as the LLMs explained. The color encoding is merely inferred, as users often explicitly specify the color encoding such as *"color by region"*.
- **Design (96%).** The design component is the most frequently inferred specification, due to users rarely providing explicit design preferences in their utterances. The LLMs typically infer the design properties based on their own preferences. Notably, the Gemini model prefers to use default settings for mark style (814), axis title (567), and axis format (384), which often misalign with the target visualization. When the utterance involves specific analytic intent, the model may also infer the design properties based on the context, such as *"show me the distribution of average sales by region"* is inferred as a histogram with a binning design and a proper axis format.

3.3 Analysis of LLM Misinterpretation

To address ***RQ2***, we evaluate the performance of LLMs on visualization generation by comparing the generated specifications against the ground truth. We adopt hard constraints for the data schema, mark, and encoding components, and soft constraints for the design component when evaluating the accuracy. The overall accuracy of a generated

visualization is defined as 1 if and only if all hard constraints strictly match the ground truth, otherwise 0. Consequently, the Gemini model achieves an overall accuracy of 47% on the 814 utterances, with 73% for data schema, 92% for mark, 69% for encoding. We then investigate how these implicit inferences may lead to LLM misinterpretation and incorrect results. The accuracy is significantly lower when the specifications are implicitly inferred, with only 53% for data schema, 85% for mark, and 71% for encoding. Through an in-depth analysis of the failure cases, we identify three common misinterpretation patterns.

Limited Expression of Visual Intent. Text prompts are inherently inflexible and inconvenient to express users' visual intent. First, users usually use vague terms like *"relation"* or *"associate"*, rather than explicitly stating chart types like *"line"* or *"scatter"*. This typically yields much lower fidelity for marks, as LLMs might fail to capture users' nuanced intents. For example, *"relationship between release year and average production budget"* leads to a bar chart instead of the expected line chart. Second, when dealing with encodings, text prompts often lack axis-specific details that clearly define the visual mappings and layout. A common misinterpretation by LLMs is the incorrect assignment of data attributes to the x-axis or y-axis, unless explicitly specified or certain phrases like *"horizontal"* or *"vertical"* are used. Additionally, when users' utterances involve a third encoding channel without providing explicit instructions, LLMs frequently struggle to infer the correct encoding type like color or size, especially in scatterplots or bubble charts. In summary, text prompts often fail to capture the full range of users' visual intent due to linguistic ambiguities and the difficulty of describing precise visual relationships through text alone.

Inadequate Guidance of LLM Behavior. It is observed that, even when visualization specifications are explicitly stated in the utterances, LLMs still struggle to generate accurate visualizations. In certain cases, the absence of specific keywords constantly leads to LLMs' incorrect inferences. For example, keywords like *"across"*, *"over"*, and *"by"* are very likely to guide LLMs to the correct encoding choices, while other terms like *"between"* or *"versus"* are less effective. However, users are often unaware of which keywords are essential for guiding LLMs' behavior, leading to misinterpretation. The situation is exacerbated when dealing with complex utterances that involve multiple encoding channels or uncommon designs. Even when users provide specific and detailed instructions, LLMs still struggle to generate accurate specifications. For example, we find that LLMs keep failing to create multi-view visualizations, such as small multiples or concatenated views, unless users explicitly mention relevant terms (*e.g.*, *"separate"*, *"split"*). A typical case is the misinterpretation of *"the average of Production Budget categorized by Creative Type and parameterized by Content Rating"*, where LLMs always generate a single stacked bar chart instead of a faceted view. These observations indicate that users' lack of knowledge in formulating effective text prompts frequently hinders LLMs from interpreting and fulfilling their requirements.

Misaligned Human-LLM Design Preferences. Visualization authoring relies on specific design knowledge drawn from established best practices. However, users' design preferences are often misaligned with LLMs' design choices learned from their training data. In our analysis, we find that LLMs tend to overwhelmingly favor certain default settings or common design patterns, leading to unexpected or inappropriate results. This misalignment is particularly evident in the misinterpretation of stylistic aspects like axis formats, color schemes, or chart titles. As users rarely provide explicit instructions for these visualization properties, LLMs resort to their preconceived notions, which may deviate from users' expectations, or sometimes decrease the readability or aesthetics of the visualization. For example, LLMs often generate scatterplots with hollow circles instead of filled ones, or use a sequential color scheme instead of a diverging one, which may not be suitable for the given data context or analytic tasks. Another common issue is the handling of time units, where LLMs frequently fail to infer the correct time granularity for the x-axis, causing visual clutter. These discrepancies between users' design preferences and LLMs' learned behaviors demand extensive customization operations and iterative enhancements, which can be frustrating and time-consuming.

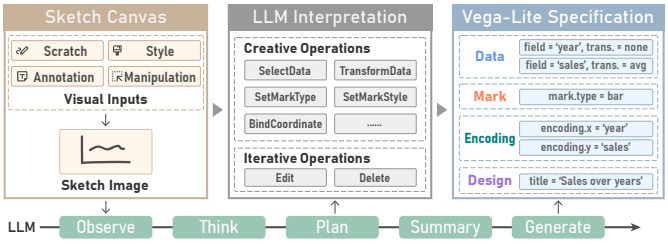

Fig. 4: Our prompting framework instructs the LLM to interpret visual prompts to visualization specifications step by step.

4 Leveraging Visual Prompt

Informed by our findings from the empirical study, we design a multimodal prompting framework that incorporates visual prompts to instruct LLMs for visualization authoring tasks.

4.1 Design Considerations

To address the limitations of text prompt (Sec. 3.3), our prompting framework is designed following three key design considerations:

C1 Expressiveness: The visual prompting should enable users to express their visualization intent in a flexible and comprehensive manner, addressing the *limited expression of visual intent* present in text prompting approaches.

C2 Structured Interpretation: Visual prompts should be systematically captured and translated into precise specification constraints that LLMs can accurately interpret, overcoming the *inadequate guidance* issues and ambiguities inherent in text prompts.

C3 Interaction Intuitiveness: Visual interactions should leverage users' natural understanding of visualization creation, enabling iterative refinement while bridging the *design preference bias between human and LLMs* that often requires extensive customization in the text prompting approaches.

4.2 Visual Input Actions

To support creating expressive visual prompts with flexible interactions (**C1**), we propose four fundamental visual input actions that can be performed on a sketch canvas. These actions are inspired by a comprehensive literature review of existing visualization authoring tools [2, 28, 39], which highlights common user interactions in visualization authoring tasks. As such, users can freely create expressive visual prompts in a sketch canvas environment using these actions:

- **Scratch**: Users can sketch and layout visual elements on the canvas to represent their desired visualization. This action allows users to express their visual intent through freehand drawing, which can be interpreted as a specific mark type or layout.
- **Style**: Users can apply visual styles to the marks, such as color, size, and shape, to convey their design or encoding preferences.
- **Annotation**: Users can annotate to the sketch visualization to provide additional context or information about specific elements.
- **Manipulation**: Users can perform direct manipulation to select and modify specific elements of existing visualization, such as resizing or repositioning marks, to refine their design.

4.3 Visualization Specification Operations

To support LLMs in understanding visual input actions and accurately interpreting user intent (**C2**), we summarize the potential user intents as two categories of operations: Creative Operations and Iterative Operations. The creative operations are derived from the specification components identified in our corpus analysis and are designed to be machine-interpretable. Each operation represents the user intent on a specific aspect of the visualization specification. Meanwhile, the iterative operations are designed to refine and enhance existing visualizations (**C3**). To simplify the presentation, we include only the most common operations in this paper, while the complete list of operations is provided in the supplementary material.

Fig. 5: The interface of VisPilot includes four components: (A) Chat Interface, (B) Free-drawing Canvas, (C) Design Panel, and (D) Authoring Flow.

Creative Operations. The creative operations aim to create a new visualization from scratch, allowing users to express their visual intent covering the entire specification space.

- **SelectData** specifies the data attributes to be visualized. Users can indicate their selection of attributes by T annotate keywords in the chart title, legend, or axis labels. This operation represents constraints on the field properties in the data schema specification.
- **TransformData** specifies the data transformations to be applied to the selected data attributes. Users can express the transformation requirements through explicit T annotations along with data attributes or visual examples, for instance, sketching a descending bar chart to indicate sorting. This constrains a series of properties in the data schema specification, including aggregate, sort, and transform, etc.
- **SetMarkType** specifies the type of mark to be used in the visualization. Users can sketch desired visual marks on the canvas, which presents constraint on the type property in the mark specification.
- **SetMarkStyle** specifies the visual styles to be applied to the marks. Users can apply styles to the marks, such as color, size, and opacity, to convey their design or encoding preferences. This constrains on either properties in the design specification such as mark.fill, or properties in the encoding specification such as encoding.color.
- **BindCoordinate** specifies the mapping of data attributes to coordinate axes. Users can sketch the coordinate axes with data attributes T labeled on them to indicate their intents. This typically constrains properties like x.field, y.field in the encoding specification.
- **Layout** specifies the layout of the visualization views. Users can sketch the layout of the visualization views on the canvas, such as arranging multiple charts in a grid or juxtaposition. This constrains properties such as column.field or facet in the encoding specification.

Iterative Operations. The iterative operations refine and enhance an existing visualization through progressive refinement.

- **Edit** modifies existing visualization specifications to refine the visualization. Users can select elements to highlight their target with visual expressions, such as changing the color of marks or modifying the axis title. This instructs the model to edit the corresponding visual elements on the basis of the existing visualization specification.
- **Delete** removes specific elements or properties from the existing visualization. Users can select elements to remove or properties to disable. This sets specific properties to null/false or removes them entirely from the Vega-Lite specification.

4.4 Interpretation of Visual Prompts

To instruct LLMs to accurately interpret visual prompts, we propose a novel multimodal prompting framework that systematically guides the model through visual input interpretation and visualization generation. Our framework structures the LLM reasoning process through five sequential steps, as illustrated in Figure 4.

The model reasoning starts with **Observation**, where the model receives visual inputs and observes user actions on the sketch canvas. The model is required to report its observations with specific location information (*e.g.*, *"the user annotated 'year' on the x-axis"*) to ensure it accurately captures user actions in a visual context. Next, the model proceeds to **Thinking**, where it infers the user's underlying visualization intent based on the observed visual input context and the conversation history, identifying which specification components (data, mark, encoding, etc.) the user is trying to express. After that, the model enters the **Planning** phase, where it maps the inferred intent to concrete specification operations (SelectData, SetMarkType, BindCoordinate, etc.) and generates a list of operations to be performed. Subsequently, the model generates a **Summary** of its understanding and planned operations, explicitly stating its interpretation of the user's intent and the operations it will perform to fulfill that intent. Finally, the reasoning process ends with the **Generation** phase, where the model produces a complete Vega-Lite specification that implements all the visualization requirements expressed through the visual prompts.

5 The VisPilot System

To evaluate the feasibility of multimodal prompting for visualization authoring, we design VisPilot as a proof-of-concept system. Figure 5 shows an overview of its interface including four main components.

Chat Interface. The chat interface comprises a data table view and a chat view (Fig. 5-A). Users can upload a dataset to start the conversation with the system. The dataset is displayed in the data table view and automatically sent to the LLM. Below the data table, the

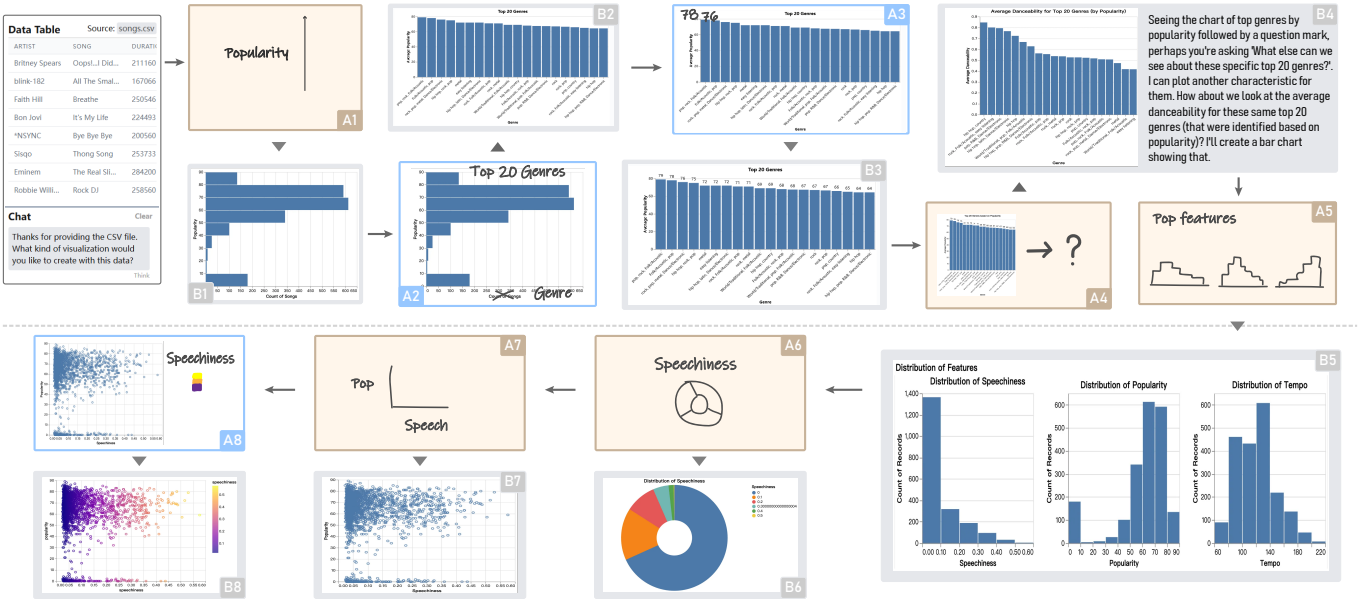

Fig. 6: The use case of VisPilot for data exploration. A user explores a dataset containing information about the top tracks on Spotify through an iterative process of sketching (A1–A8) and visualization generation (B1–B8).

LLM will initially respond to the dataset and ask the user for a specific visualization task. In the following conversation, users can instruct the LLM to generate visualizations by natural language in this chat view. As we employ the prompting framework presented in Sec. 4.4, the LLM commonly generates long responses containing its thinking process and visualization specification code. We hide these long responses and only show the summary part of responses to avoid overwhelming users with too much information.

Free-drawing Canvas. The free-drawing canvas allows users to create visual prompts for the LLM (Fig. 5-B). Users can use mouse, touch, or pen as input devices to draw on the canvas. Refer to the prompting framework (Sec. 4.4), we provide a set of widgets to help users create visual prompts for the LLM. The pen, shape, axis, and text widgets are used to create sketches of visualizations, while the two selection widgets are used to select the elements drawn by users or generated by the LLM for further manipulation. Once the user clicks on the Ask VisPilot button, the system will send the sketch image as a visual prompt to the LLM.

Design Panel. The design panel presents a configuration panel and a design idea panel (Fig. 5-C). The configuration panel allows users to configure the style properties of selected elements in the canvas. The design ideas panel displays the design ideas generated by the LLM based on user instructions, including the generated Vega-Lite specification code and the corresponding visualizations. Users can add the satisfied visualizations to the canvas or ask the LLM to generate alternative design ideas.

Authoring Flow. The authoring flow of VisPilot is shown in Fig. 5-D. It shows the steps of the authoring process made by the user. All user interactions and LLM responses are recorded in the authoring flow. The interactions for text prompting and visual prompting are shown in gray and blue colors, respectively. Each visualization generated by the LLM is shown below the timeline, presenting the process of how the user completed the authoring task.

VisPilot is implemented as a web application using React and Vega-Lite, with generative capability supported by APIs from commercial LLMs, such as GPT-4o and Gemini 2.0. The system is accessible through desktop and tablet devices, allowing users to input multimodal prompts using the mouse, touch, pen, or keyboard.

6 Case Studies

To demonstrate the effectiveness of VisPilot, we present two cases that showcase how users can leverage multimodal prompts to complete visualization authoring (Sec. 6.1) and data exploration (Sec. 6.2) tasks.

6.1 Case 1: Visualization Authoring

The first case study follows Alice, a data journalist, in her authoring process of a visualization showing how different categories of consumer goods contribute to inflation over time (Fig. 1). The process begins with Alice uploading a dataset containing the Consumer Price Index (CPI) data from 2003 to 2021. In the data table view, she can see the dataset including the CPI values for various categories of goods and services, such as food, education, and transportation. Afterwards, she has an idea of creating a streamgraph-style area chart to visualize the contribution of different categories to the overall inflation rate, which she sketches on the free-drawing canvas (A1). VisPilot then generates a visualization based on her sketch, displaying the contribution of different categories to the overall inflation rate over time (B1).

Alice is satisfied with the visual form of the generated chart but does not like the default color scheme, which she finds visually complex. To address this, she modifies her sketch by drawing a colored example on the right side of the chart, indicating her preference for a gradient blue color palette (A2). VisPilot responds by implementing a sequential blue color scheme in the chart (B2). To further enhance the readability of the chart, Alice refines the color scheme by reversing example colors, and annotates titles on the sketch (A3). After the system generates the updated chart with the new color scheme and titles (B3), Alice is pleased with the overall design but wants to make some final adjustments directly on the visualization.

She uses the selection tool to select the legend and draws an arrow pointing to her desired position (A4), prompting the system to reposition the legend to the top-right area of the plot (B4). She further simplifies the design by crossing out the axis titles (A5), which the system removes from the chart (B5). As a complementary step, Alice decides to add a statement under the chart title to provide context for the visualization (A6). VisPilot interprets her prompts and adds the statement as the subtitle of the chart (B6). Finally, Alice is satisfied with the final design and saves the visualization. This case study demonstrates that our prompting framework is effective in guiding the LLM to generate visualizations that precisely align with user preferences.

6.2 Case 2: Data Exploration

The second case demonstrates how VisPilot can assist users in data exploration tasks which involve continuous progress of data analysis and visualization authoring (Fig. 6). This case study follows Bob, a data analyst, who is interested in exploring a dataset containing information about the top 2000 tracks on Spotify from 2000 to 2019.

Bob is interested in popular music and wants to explore the dataset to find out which genres are the most popular and how they relate to other features of the songs. After uploading the dataset and examining the data table, he decides to start his exploration by viewing the distribution of song popularity. He sketches a coordinate axis with the annotation "popularity" beside the axis (A1), and the system responds with a histogram showing the distribution of song popularity (B1). Bob is intrigued by the highly popular tracks and modifies the visualization by changing the x-axis title to "genre" and annotating the title "Top 20 Genres" (A2). The system generates a ranked bar chart of the top 20 genres by mean popularity (B2). To enhance the visualization's informativeness, Bob indicates his interest in seeing specific values (A3), prompting the system to add popularity value annotations to the bars (B3). Seeking further insights, Bob wants the LLM to recommend another visualization derived from the current one. He draws an arrow pointing to a question mark beside the chart (A4), which leads the system to visualize the average danceability of the top 20 genres (B4).

Bob's curiosity then leads him to explore more features that might be related to the popularity of songs. He sketches three hismograms and annotates "popularity-related features" on the top (A5). The system generates three histograms showing the distributions of danceability, energy, and tempo (B5). Particularly interested in the speechiness attribute, he sketches a donut chart labeled "speechiness" (A6), which the system renders to reveal a pattern where songs with higher speechiness values were less common in the dataset (B6).

For his final analysis, Bob draws a coordinate axis labeled with "speechiness" and "popularity" (A7), prompting the system to generate a scatter plot showing their relationship (B7). To enhance the visualization's clarity, he adds a hip-hop-styled color legend for speechiness (A8), which the system incorporates into a color-encoded visualization of the relationship between these variables (B8).

7 User Study

We conducted a controlled user study to evaluate the effectiveness of multimodal prompting for visualization authoring tasks.

7.1 Methodology

Our study followed a within-subject design to account for the individual differences among participants. VisPilot was used as the technology probe to explore the potential of multimodal prompting for visualization authoring. We aim to evaluate the multimodal prompting approach from three aspects: (1) the accuracy and efficiency of creating visualizations, (2) the usability and user experience of the system, and (3) the user behavior and interaction patterns in different prompting conditions.

Conditions. The study was conducted in two prompting conditions:

- **Text Condition.** In the text prompting condition, users can interact with the system only through the chat interface, which provides similar user experience to common LLM chat interfaces (*e.g.*, ChatGPT). Users can type their natural language utterances, and the system responds with chat messages and visualizations.
- **Multimodal Condition.** In the multimodal prompting condition, users can interact with the system without restrictions. Comparing to the text condition, users have additional capabilities to create visual prompts from scratch or existing visualizations. They are free to use text or multimodal prompts to convey their intents, and the system respond with chat messages and visualizations as the text condition.

The two conditions both used the gemini-2.0-flash-001 model as the LLM backend, along with the same configuration and system prompt.

Participants. We recruited 10 participants, who were all familiar with the use of LLMs and had used them for creating data visualizations. Two of them worked in a fintech company, who need to create visualizations in their daily work. The other eight participants were graduate students from a local university, who commonly need to create visualizations for data analysis and research.

Tasks. We designed a set of replication tasks in the study. There were four groups of visualizations for the participants to create (see Fig. 7), including two simple-level groups (G1, G2), and two complex-level

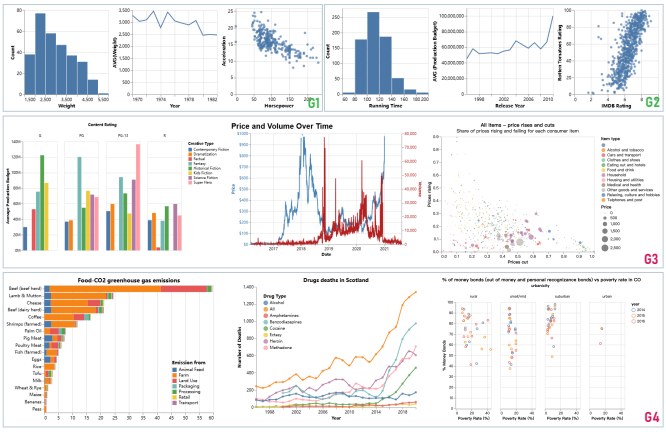

Fig. 7: The target visualizations of tasks (G1, G2, G3, G4).

groups (G3, G4). Each group was designed to contain three commonly used types of visualizations involving bar chart, line chart, and scatter plot. Participants were asked to replicate one simple and one complex group of visualizations in each condition (12 trials in total). They were asked to create visualizations as similar as possible to the targets, including the data schema, marks, visual encodings, and design details.

Procedure. We conducted a 90-minute session with each participant. After obtaining informed consent, participants completed a pre-study questionnaire about their experience with data visualization tools and LLM-based systems. We then provided a 10-minute introduction to VisPilot, including a demonstration of its features in both text and multimodal conditions.

The main study consisted of four trials where participants replicated visualization groups using VisPilot. We counterbalanced the order of conditions and task complexity across participants to mitigate learning effects. Each participant experienced both conditions (text-only and multimodal) and both complexity levels (simple and complex). For each trial, participants were given the target visualization and instructed to recreate it as accurately as possible within a 10-minute time limit. We encouraged participants to think aloud during the process.

After completing all trials, participants filled out a post-study questionnaire to evaluate the system usability and cognitive load. We concluded with a semi-structured interview to gather qualitative feedback about their experiences using different prompting modalities. All sessions were recorded for later analysis with screen recordings. Each participant was compensated with $20 for their time and effort.

Metrics. We measured two prompting conditions in terms of quantitative and qualitative metrics. The quantitative metrics evaluated the task performance including task efficiency and accuracy. For task efficiency, we recorded the *completion time* at two time points, including the first-time creation of the visualization and the last-time generation by the LLM, which represent the time for completing creation and iteration tasks, respectively. For task accuracy, we recorded the *number of mismatches* at two time points as above, which represents the number of mismatched visualization properties. Qualitative metrics evaluated system usability and user experience through 7-point Likert scale questionnaires, collected from post-study interviews.

7.2 Results

We analyzed the data collected from the user study, including task performance metrics, qualitative feedback, and user behavior patterns.

7.2.1 Quantitative Results

Overall, the multimodal condition achieved higher task accuracy without significantly affecting task efficiency compared to the text condition. We conducted a series of paired t-tests to compare the two conditions ($\alpha = 0.05$). The details of the results are as follows:

Task Efficiency. The multimodal condition did not significantly differ from the text condition in terms of completion time for the first-time creation of visualizations (Simple: $t = 1.13, p = .265$; Complex: $t =$

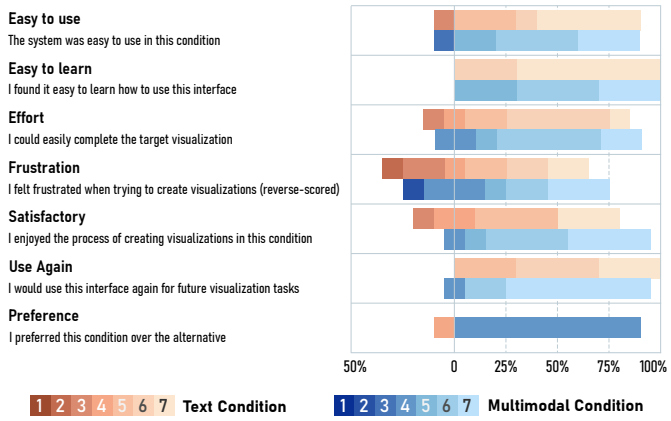

Fig. 8: The questionnaire and qualitative results, where rating from 1 to 7 represents *strongly disagree* to *strongly agree*.

$1.03, p = .308$) and the last-time generation by the LLM (Simple: $t = -0.11, p = .913$; Complex: $t = 1.04, p = .304$). This suggests that, even typing text prompts was generally faster than drawing visual prompts, the added workload did not significantly affect the overall task efficiency. During the study sessions, we constantly observed that, in the text condition, participants tended to spend more time on thinking and formulating the prompts than on the actual interaction with the system, while vice versa in the multimodal condition. This can be ascribed to the intuitiveness and directness of visual prompts in the multimodal condition, which allowed participants to quickly iterate on their visualizations, thereby complementing the additional time spent on drawing and manipulating visual elements.

Task Accuracy. The multimodal condition achieved fewer mismatches in total compared to the text condition for the first-time creation of visualizations (Simple: $\Sigma = 37 < 44$; Complex: $\Sigma = 130 < 155$) and the last-time generation by the LLM (Simple: $\Sigma = 0 < 2$; Complex: $\Sigma = 6 < 14$). Generally, multimodal prompts helped participants obtain visualizations that were more aligned with their expectations. We noticed that for subtle design properties such as axis ticks/formats, color schemes, and legend placements/orientations, using visual prompts was more precise and user-friendly than pure texts. This demonstrates the superiority of multimodal prompting in conveying detailed and nuanced visual intents and reducing users' cognitive load, especially for complex visualizations that exhibit uncommon encoding choices or require specific spatial and layout considerations.

7.2.2 System Usability

The usability results are summarized in Fig. 8. Overall, participants appreciated the usability of the multimodal condition for creating visualizations. We analyze the results from the following dimensions:

Easy to Use and Learn. The multimodal condition achieved a comparable score to the text condition in terms of ease of use ($\mu = 5.8, \sigma = 1.2$) and learnability ($\mu = 6.0, \sigma = 0.8$). The reason why the multimodal condition did not score higher was that, text prompts were inherently intuitive and easy to use for participants who were already familiar with LLMs. Nevertheless, it did not require a steep learning curve to use our proposed visual input actions, as stated by most participants. Once they became accustomed to the interface, they found it *"natural and intuitive"* (P3), just like *"drawing on a whiteboard"* (P5). Also, some participants mentioned that the multimodal condition would be *"more accessible to people not familiar with LLMs"* (P2) and *"even more useful with a tablet for students to learn about visualizations"* (P2).

Effort and Frustration. The multimodal condition scored higher in terms of effort ($\mu = 5.7, \sigma = 1.1$) and frustration ($\mu = 5.2, \sigma = 1.7$) compared to the text condition. We observed that participants struggled with the text condition when creating visualizations that required spatial and layout considerations. As stated by P8, *"I could not move the legend to the bottom-right corner inside the chart, no matter how hard I tried to rephrase my text prompts"*. This significantly increased their effort and frustration levels. In contrast, the multimodal condition allowed participants to directly add an arrow on the canvas to indicate the desired position, which was more *"effective and convenient"* (P9). However, P6, who was experienced with the Vega-Lite grammar, found the multimodal condition *"less efficient for simple tasks"*, as he could proficiently write accurate instructions rather than tediously drawing them. This suggests that different users may have different preferences for the two prompting modalities based on their domain expertise.

Satisfaction and Real-world Application. The multimodal condition was more favored than the text condition in terms of satisfaction ($\mu = 6.1, \sigma = 1.0$) and willingness to use again ($\mu = 6.5, \sigma = 1.0$). For example, P2 remarked, *"visual interactions were more engaging and fun to use, making me feel like I was designing visualizations rather than just typing text prompts"*. They also suggested several features to improve the multimodal interface, such as *"supporting predefined sketch templates"* (P9) and *"enabling more precise selection and control of visual elements"* (P5). Additionally, most participants were willing to integrate visual prompts into their daily workflow for creating visualizations. They believed that visual inputs could be *"a great complement to existing text-based visualization authoring tools"* (P10) and could potentially be applied to various scenarios, such as *"visualization education"* (P2), *"collaborative design sessions"* (P6), and *"exploratory data analysis"* (P1).

Preference. Most participants (9/10) expressed a preference for the multimodal condition over the text condition. While text prompts are concise and intuitive for simple user requirements, participants acknowledged their inherent limitations in conveying nuanced visual intents, especially for complex visualizations and detailed visual changes. Also, as stated by P9, the multimodal condition was particularly useful for *"creative scenarios where the desired outcome might be difficult to articulate with words alone"*. Meanwhile, some participants (P3, P8) wished to combine visual input with text input to *"make the most of each other's advantages"* (P3). Regarding the only one participant (P7) who preferred the text condition, he explained that, *"for me, text prompts are more efficient to express my intents, for example, typing the word 'sort' is far more convenient compared to drawing multiple bars with lengths that decrease in size"*. We attribute this to different user habits, reinforcing the importance of providing multiple prompting modalities to cater for various user needs and preferences.

7.2.3 User Behaviors and Interaction Patterns

We observed notable user behavior patterns across the two prompting conditions, beyond the quantitative and qualitative metrics.

Text Prompting Strategies. In the text condition, participants often engaged in iterative prompt refinement: when initial outputs did not match their intent, they would rephrase, simplify, or break down their requests into smaller steps. For instance, P8 described a process of "trial and error with wording" to achieve the desired legend placement, but still found it difficult to control spatial details.

Multimodal Iteration Patterns. In the multimodal condition, participants frequently adopted a "sketch-and-adjust" workflow: they would first sketch the overall layout visually, then use direct manipulation to fine-tune elements, and occasionally supplement with text for precise specifications. This allowed for rapid, incremental adjustments and immediate feedback, which participants found intuitive and engaging.

Cognitive Load and Focus. We observed that in the text condition, participants spent more time planning and formulating prompts, while in the multimodal condition, their attention shifted to manipulating visual elements and interpreting system feedback. Several participants (e.g., P3, P5) noted that visual interaction *"felt more like designing"* and reduced the need to mentally translate visual ideas into words.

Emergent Best Practices. Some participants developed hybrid strategies, such as using visual prompts for layout and text for data mapping, sketching with text annotation, suggesting the value of combining modalities. Others suggested features like "importing external image" and "template-based sketching" to further streamline the workflow.

These observations reveal that multimodal prompting not only improves expressiveness and efficiency for complex tasks, but also supports more natural and flexible authoring behaviors.

8 Discussion

The results of the case studies and user evaluation provide valuable insights into the design of multimodal prompting approaches for visualization authoring tasks. We summarize the implications as follows.

8.1 Strengths and Limitations of Prompting Modalities

Our study findings highlights how each modality contributes to the overall authoring experience and where their boundaries lie.

Visual prompts facilitate LLMs in understanding user intent and improve generation accuracy. Our study showed that visual prompts significantly enhance how LLMs interpret user intentions, especially for complex visualization requirements. Sketches that conveyed chart types, element positioning, and visual encoding relationships led to visualizations that more closely matched user intentions compared to text-only prompts. Visual inputs provided direct spatial representations that text often struggled to communicate clearly, reducing ambiguity and eliminating the need for LLMs to make multiple inferences that could lead to misinterpretations. This finding demonstrates the potential of visual prompts to improve the accuracy of LLM-generated visualizations, particularly in scenarios where users need to convey intricate visual intent or spatial relationships.

Visual inputs reduce the effort for expressing sophisticated visualization demands. Our study revealed that visual prompts significantly reduce the cognitive burden of articulating complex visualization requirements. Participants utilized visual prompts more intuitively and efficiently than text prompts when communicating spatial relationships, layout preferences, and design modifications. This was especially notable for tasks requiring precise element positioning such as legends or annotations, where a simple visual indicator achieved what would otherwise necessitate multiple textual exchanges. The multimodal approach allowed users to express intent through the most natural modality for each visualization aspect, reducing the overall effort required.

Text prompts excel in parameter specification and conceptual guidance. While visual prompts suit spatial relationships, text offers greater efficacy for precise parameter values and high-level analytical goals. Participants exhibited increased efficiency when textually specifying numeric parameters (*e.g.*, "set opacity to 0.7" or "use a logarithmic scale for the y-axis") rather than drawing visual representation of these concepts. Similarly, text proved more effective for communicating abstract visualization objectives like "show the correlation between variables" or "highlight the outliers". Future systems should integrate text prompts for precise specifications with visual inputs for spatial and design elements, establishing a complementary multimodal framework.

Direct manipulation enables efficient iterative refinement. Our study demonstrated that direct manipulation of visualization elements significantly enhanced the refinement process. Participants strongly preferred this iterative design paradigm, which enabled them to progressively refine existing visualizations through visual interactions rather than verbal descriptions. This was particularly evident in tasks requiring fine-grained adjustments such as repositioning legends, modifying mark properties, or altering axis scales, as visual actions like drawing arrows for movement or crossing out elements for removal were far more efficient than composing detailed textual instructions. Therefore, it is beneficial to integrate direct manipulation operations along with multimodal inputs, which can provide seamless transitions between creation and iteration phases with appropriate visual guidance.

Lengthy instructions lead to LLM misinterpretation regardless of prompt modality. We observed that complex instructions with multiple requirements frequently resulted in partial implementation by the LLM, with certain aspects prioritized while others were overlooked. This phenomenon was pronounced when instructions contained conflicting or ambiguous specifications. For example, detailed visualization requirements with multiple design constraints often led to selective implementation. Similarly, visual prompts with excessive annotations overwhelmed the LLM's processing capacity. Our findings indicate that concise, focused instructions produce superior outcomes, and complex requirements should better be decomposed into sequential interactions rather than consolidated within a single prompt.

8.2 Opportunities in Multimodal Prompting

We further explore the broader opportunities that multimodal prompting brings to visualization authoring and implications for future research.

Balance between text and visual prompts. Our study highlighted the importance of finding an optimal balance between text and visual prompts for effective visualization authoring. We observed that participants typically adopted a complementary approach, leveraging each modality's inherent strengths rather than relying exclusively on a single one. Text prompts were utilized predominantly for data transformations, parameter specifications, and high-level objectives, while visual prompts proved superior for communicating nuanced visual details involving layout or spatial relationships. This complementary pattern emerged consistently across participants and tasks, which inspires future work to facilitate fluid transitions between modalities, rather than treating them as discrete interaction paradigms and optimizing each modality of prompts [6] in an isolated manner.

Visual context can improve LLMs' reasoning abilities. During our experiments, LLMs exhibited enhanced reasoning capabilities for visualization tasks when provided with rich visual context. This could be attributed to the integrated visual prompts that potentially improved LLMs' comprehension of current visualization states and users' modification intentions. Specifically, when users made incremental changes to existing visualizations, LLMs could successfully interpret visual context and applied appropriate transformations. However, we also noticed that LLMs occasionally misinterpreted ambiguous visual cues, especially when they were arbitrary scratches or using obscure colors. Therefore, future research is needed to make LLMs' visual reasoning capabilities more robust for visualization-specific tasks through model training, fine-tuning or prompt engineering techniques.

Multimodal prompt design can inspire brainstorming. Some participants commented that multimodal prompting facilitated creative visualization exploration and ideation. Unlike text-only approaches that require users to have a clear visualization concept from the start, visual prompts allowed for more exploratory and iterative design processes. Participants frequently started with rough sketches to express general concepts, then progressively refined them through a combination of visual and textual inputs. Such a brainstorming paradigm was particularly well-suited for complex visualization tasks, where the optimal design was not immediately apparent. Moreover, the ability to quickly sketch alternative designs and receive immediate feedback from the system encouraged participants to explore diverse visualization options rather than settling for their first idea. Future work could further enhance this procedure by providing features for quick sketching, visual variations, and easy comparison of alternative designs. Our findings can also inform the application of multimodal prompting for other creative domains, such as writing [17] and image painting [11].

9 Conclusion

In this paper, we present VisPilot, a novel approach that addresses the fundamental limitations of text-only prompts for visualization authoring with large language models. Through our empirical study, we identify three key challenges in text-only prompting: limited expression of visual intent, inadequate guidance of LLM behavior, and misaligned human-LLM design preferences. Our multimodal prompting framework directly addresses these limitations by enabling users to express their visual intent through complementary visual and textual inputs. We develop the VisPilot system and conduct two case studies and a formal user study to validate its effectiveness. The results of studies demonstrate that multimodal prompting outperforms text-only prompting in terms of accuracy in task completion and user satisfaction. These findings provide strong evidence that incorporating visual prompts can effectively enhance the LLM-based visualization authoring workflow. As multimodal large language models continue to evolve, we believe the paradigm of multimodal prompting will become increasingly important for visualization authoring scenarios, making data visualization more accessible while preserving the expressivity and control that visualization authors require.

REFERENCES

[1] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011. doi: 10.1109/TVCG.2011.185 1

[2] J. Chen, J. Wu, J. Guo, V. Mohanty, X. Li, J. P. Ono, W. He, L. Ren, and D. Liu. Interchat: Enhancing generative visual analytics using multimodal interactions. *arXiv preprint*, abs/2503.04110, 2025. doi: 10.48550/arXiv.2503.04110 1, 2, 4

[3] N. Chen, Y. Zhang, J. Xu, K. Ren, and Y. Yang. VisEval: A benchmark for data visualization in the era of large language models. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):1301–1311, 2025. doi: 10.1109/TVCG.2024.3456320 2

[4] V. Dibia. LIDA: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models. *arXiv preprint*, 2023. doi: 10.48550/arXiv.2303.02927 1, 2

[5] Y. Feng, X. Wang, B. Pan, K. K. Wong, Y. Ren, S. Liu, Z. Yan, Y. Ma, H. Qu, and W. Chen. XNLI: Explaining and diagnosing NLI-based visual data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 30(7):3813–3827, 2024. doi: 10.1109/TVCG.2023.3240003 1

[6] Y. Feng, X. Wang, K. Wong, S. Wang, Y. Lu, M. Zhu, B. Wang, and W. Chen. PromptMagician: Interactive prompt engineering for text-to-image creation. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):295–305, 2024. doi: 10.1109/TVCG.2023.3327168 9

[7] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. DataTone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST 2015, Charlotte, NC, USA, November 8-11, 2015*, pp. 489–500. ACM, 2015. doi: 10.1145/2807442.2807478 2

[8] T. Gupta and A. Kembhavi. Visual programming: Compositional visual reasoning without training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 14953–14962. IEEE, 2023. doi: 10.1109/CVPR52729.2023.01436 2

[9] D. Han, H. Zhu, W. Chen, J. Pan, R. Chen, X. Wang, Z. Wen, L. Weng, M. Zhu, Y. Wu, and R. Westermann. Nuwa: An authoring tool for graph visualizations. In *Proceedings of the 17th IEEE Pacific Visualization Conference, PacificVis 2024, Tokyo, Japan, April 23-26, 2024*, pp. 142–151. IEEE, 2024. doi: 10.1109/PACIFICVIS60374.2024.00024 1

[10] 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*, 2023. doi: 10.48550/arXiv.2311.16483 1, 2

[11] T. Hang, S. Gu, D. Chen, X. Gen, and B. Guo. CCA: collaborative competitive agents for image editing. *Frontiers of Computer Science*, 19(11):1911367, 2025. doi: 10.1007/s11704-025-41244-0 9

[12] J. Hao, Z. Liang, C. Li, Y. Luo, and W. Zeng. Visltr: Visualization-in-the-loop table reasoning. *arXiv preprint*, abs/2406.03753, 2024. doi: 10.48550/arXiv.2406.03753 2

[13] J. Hao, M. Yang, Q. Shi, Y. Jiang, G. Zhang, and W. Zeng. FinFlier: Automating graphical overlays for financial visualizations with knowledge-grounding large language model. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–17, 2024. doi: 10.1109/TVCG.2024.3514138 2

[14] S. Huang, S. Mamidanna, S. Jangam, Y. Zhou, and L. H. Gilpin. Can large language models explain themselves? a study of llm-generated self-explanations. *arXiv preprint arXiv:2310.11207*, 2023. doi: 10.48550/arXiv.2310.11207 3

[15] J. Kassel and M. Rohs. Valletto: A multimodal interface for ubiquitous visual analytics. In *Proceedings of the Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, Montreal, QC, Canada, April 21-26, 2018*. ACM, 2018. doi: 10.1145/3170427.3188445 2

[16] G. Li, X. Wang, G. Aodeng, S. Zheng, Y. Zhang, C. Ou, S. Wang, and C. H. Liu. Visualization generation with large language models: An evaluation. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2401.11255 2

[17] Y. Liu, Z. Wen, L. Weng, O. Woodman, Y. Yang, and W. Chen. Sprout: an interactive authoring tool for generating programming tutorials with the visualization of large language models. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–15, 2024. doi: 10.1109/TVCG.2024.3410523 9

[18] J. Lu, R. Gan, D. Zhang, X. Wu, Z. Wu, R. Sun, J. Zhang, P. Zhang, and Y. Song. Lyrics: Boosting fine-grained language-vision alignment and comprehension via semantic-aware visual objects. *arXiv preprint*, abs/2312.05278, 2023. doi: 10.48550/arXiv.2312.05278 2

[19] T. Luo, C. Huang, L. Shen, B. Li, S. Shen, W. Zeng, N. Tang, and Y. Luo. nvBench 2.0: A benchmark for natural language to visualization under ambiguity. *arXiv preprint*, 2025. doi: 10.48550/arXiv.2503.12880 1

[20] S. L'Yi, A. van den Brandt, E. Adams, H. N. Nguyen, and N. Gehlenborg. Learnable and expressive visualization authoring through blended interfaces. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):459–469, 2025. doi: 10.1109/TVCG.2024.3456598 1

[21] S. L'Yi, Q. Wang, F. Lekschas, and N. Gehlenborg. Gosling: A grammar-based toolkit for scalable and interactive genomics data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):140–150, 2022. doi: 10.1109/TVCG.2021.3114876 1

[22] D. Masson, S. Malacria, G. Casiez, and D. Vogel. Directgpt: A direct manipulation interface to interact with large language models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI 2024, Honolulu, HI, USA, May 11-16, 2024*, pp. 975:1–975:16. ACM, 2024. doi: 10.1145/3613904.3642462 2

[23] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):438–448, 2019. doi: 10.1109/TVCG.2018.2865240 1, 3

[24] T. Nguyen, Y. Li, U. Ojha, and Y. J. Lee. Visual instruction inversion: Image editing via image prompting. In *Proceedings of Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 1, 2

[25] B. Saket, A. Srinivasan, E. D. Ragan, and A. Endert. Evaluating interactive graphical encodings for data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 24(3):1316–1330, 2018. doi: 10.1109/TVCG.2017.2680452 2

[26] A. Saktheeswaran, A. Srinivasan, and J. T. Stasko. Touch? speech? or touch and speech? investigating multimodal interaction for visual network exploration and analysis. *IEEE Transactions on Visualization and Computer Graphics*, 26(6):2168–2179, 2020. doi: 10.1109/TVCG.2020.2970512 2

[27] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, 2017. doi: 10.1109/tvcg.2016.2599030 1, 3

[28] L. Shen, H. Li, Y. Wang, X. Xie, and H. Qu. Prompting generative AI with interaction-augmented instructions. *arXiv preprint*, 2025. doi: 10.48550/arXiv.2503.02874 1, 2, 4

[29] A. Srinivasan, B. Lee, and J. Stasko. Interweaving multimodal interaction with flexible unit visualizations for data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 27(8):3519–3533, 2021. doi: 10.1109/TVCG.2020.2978050 2

[30] A. Srinivasan, N. Nyapathy, B. Lee, S. M. Drucker, and J. T. Stasko. Collecting and characterizing natural language utterances for specifying data visualizations. In *Proceedings of CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, pp. 464:1–464:10. ACM, 2021. doi: 10.1145/3411764.3445400 2, 3

[31] A. Srinivasan and V. Setlur. Snowy: Recommending utterances for conversational visual analysis. In *Proceedings of UIST '21: The 34th Annual ACM Symposium on User Interface Software and Technology, Virtual Event, USA, October 10-14, 2021*, pp. 864–880. ACM, 2021. doi: 10.1145/3472749.3474792 1

[32] A. Srinivasan and J. T. Stasko. Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):511–521, 2018. doi: 10.1109/TVCG.2017.2745219 2

[33] 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 1, 2

[34] C. Wang, J. Thompson, and B. Lee. Data formulator: Ai-powered concept-driven visualization authoring. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):1128–1138, 2024. doi: 10.1109/TVCG.2023.3326585 1

[35] H. Wang and W. Ge. Q&A prompts: Discovering rich visual clues through mining question-answer prompts for VQA requiring diverse world knowledge. In *Proceedings of Computer Vision - ECCV 2024 - 18th European*

Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XLII, pp. 274–292. Springer, 2024. doi: 10.1007/978-3-031-72946-1_16 1, 2

[36] 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

[37] J. Wang, X. Li, C. Li, D. Peng, A. Z. Wang, Y. Gu, X. Lai, H. Zhang, X. Xu, X. Dong, Z. Lin, J. Zhou, X. Liu, and W. Chen. AVA: An automated and AI-driven intelligent visual analytics framework. *Visual Informatics*, 8(2):106–114, 2024. doi: 10.1016/j.visinf.2024.06.002 1

[38] X. Wang, Z. Wu, W. Huang, Y. Wei, Z. Huang, M. Xu, and W. Chen. VIS+AI: integrating visualization with artificial intelligence for efficient data analysis. *Frontiers of Computer Science*, 17(6):176709, 2023. doi: 10.1007/S11704-023-2691-Y 2

[39] Y. Wang, Z. Hou, L. Shen, T. Wu, J. Wang, H. Huang, H. Zhang, and D. Zhang. Towards natural language-based visualization authoring. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):1222–1232, 2023. doi: 10.1109/TVCG.2022.3209357 4

[40] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, 2016. doi: 10.1109/TVCG.2015.2467191 1

[41] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. D. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 06-11, 2017*, pp. 2648–2659. ACM, 2017. doi: 10.1145/3025453.3025768 1

[42] M. Wu, X. Cai, J. Ji, J. Li, O. Huang, G. Luo, H. Fei, G. Jiang, X. Sun, and R. Ji. ControlMLLM: Training-free visual prompt learning for multimodal large language models. In *Proceedings of Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. 2

[43] 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. *Proceedings of the ACM on Management of Data*, 2(3):115, 2024. doi: 10.1145/3654992 2

[44] S. Xiao, Y. Hou, C. Jin, and W. Zeng. Wytiwyr: A user intent-aware framework with multi-modal inputs for visualization retrieval. *Computer Graphics Forum*, 42(3):311–322, 2023. doi: 10.1111/CGF.14832 2

[45] L. Yang, Y. Wang, X. Li, X. Wang, and J. Yang. Fine-grained visual prompting. In *Proceedings of Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 2

[46] Y. Ye, J. Hao, Y. Hou, Z. Wang, S. Xiao, Y. Luo, and W. Zeng. Generative ai for visualization: State of the art and future directions. *Visual Informatics*, 8(2):43–66, 2024. doi: 10.1016/j.visinf.2024.04.003 1

[47] X. Zeng, H. Lin, Y. Ye, and W. Zeng. Advancing multimodal large language models in chart question answering with visualization-referenced instruction tuning. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):525–535, 2025. doi: 10.1109/TVCG.2024.3456159 2