SimVecVis: A Dataset for Enhancing MLLMs in Visualization Understanding

Can Liu[*] Nanyang Technological University
Chunlin Da[†] ByteDance Inc.
Xiaoxiao Long[‡] Nanjing University
Yuxiao Yang[§] Tsinghua University
Yu Zhang[¶] University of Oxford
Yong Wang[‖] Nanyang Technological University

ABSTRACT

Current multimodal large language models (MLLMs), while effective in natural image understanding, struggle with visualization understanding due to their inability to decode the data-to-visual mapping and extract structured information. To address these challenges, we propose SimVec, a novel simplified vector format that encodes chart elements such as mark type, position, and size. The effectiveness of SimVec is demonstrated by using MLLMs to reconstruct chart information from SimVec formats. Then, we build a new visualization dataset, SimVecVis, to enhance the performance of MLLMs in visualization understanding, which consists of three key dimensions: bitmap images of charts, their SimVec representations, and corresponding data-centric question-answering (QA) pairs with explanatory chain-of-thought (CoT) descriptions. We fine-tune state-of-the-art MLLMs (e.g., MiniCPM and Qwen-VL), using SimVecVis with different dataset dimensions. The experimental results show that it leads to substantial performance improvements of MLLMs with good spatial perception capabilities (e.g., MiniCPM) in data-centric QA tasks. Our dataset and source code are available at: https://github.com/VIDA-Lab/SimVecVis.

Keywords: Visualization, Multimodal LLMs, Chart QA

Index Terms: Human-centered computing—Visualization

1 INTRODUCTION

As charts have become a dominant medium for conveying data in scientific and practical contexts, manual interpretation at scale has become infeasible, calling for automated methods that can reliably understand visualizations. However, current MLLMs, originally designed for natural images, fall short in this domain due to a fundamental distinction: natural images depict real-world objects based on their visual appearance, while visualizations convey data through encoding rules that map data to visual attributes of elements such as marks, axes, and legends. Existing MLLMs are often not equipped to interpret such encoding rules, which are rarely present in natural image training data.

Visualization understanding goes beyond the mere recognition of visible content. It involves reasoning about how visual channels represent data values and inferring the underlying data in visualizations from different sources, including scanned documents and historical print media. To address the challenge of visualization understanding, we first propose SimVec, a novel **Sim**plified **Vec**tor) format to capture visual mark attributes (e.g., mark type, position, size, color), which provides a machine-readable abstraction of visualizations. Built upon it, we further construct a new visualization dataset, SimVecVis, for fine-tuning MLLMs and improving their performance of visualization understanding. SimVecVis contains 2,999 visualizations like bar charts, line charts, and area charts, and each visualization consists of three key dimensions: visualization bitmap image, SimVec representation, and data-centric QA pairs with CoT descriptions. The visualization bitmap image is commonly used for visualization understanding tasks and encodes the comprehensive information of a chart. The corresponding SimVec representation provides a more machine-friendly encoding of visualization information. The data-centric QA pairs with CoT descriptions cover chart QA tasks, like identifying the value of the tallest bar in a bar chart, and include detailed CoT reasoning descriptions, which can guide MLLMs to learn the proper strategies for **visualization understanding** as well as reasoning. We conduct extensive experiments of fine-tuning MLLMs with SimVecVis with different dataset dimensions, and the results demonstrate its effectiveness in significantly enhancing the visualization understanding performance of MLLMs with good spatial perception capabilities like MiniCPM [14]. The expressiveness of our SimVec representations is also proved via visualization information construction experiments.

The contributions of this work are summarized as follows:

- We propose a novel chart format, SimVec, for a compact and structured representation of charts.
- We construct a visualization dataset with 2,999 visualizations, SimVecVis, aiming to explicitly enhance MLLMs' visualization understanding capabilities.
- We show the expressiveness of SimVec via MLLM-based chart information reconstruction and demonstrate that fine-tuning MLLMs with SimVecVis can improve their performance in visualization understanding.

2 RELATED WORK

2.1 Artificial Intelligence for Visualization Understanding

Artificial intelligence has increasingly contributed to improving the understanding of visualizations. Early systems [7, 25, 30] were primarily rule-based, leveraging handcrafted grammars to parse chart structures and natural language. For example, Show Me [25], Voyager [33], Iris [9], and FlowSense [35] applied structured rules to support visualization querying. With the rise of deep learning, these approaches [20, 24, 27] model the relationship between visual encodings and semantics, enabling more accurate recommendations. A growing body of work focuses on understanding of existing visualizations through natural language. Key tasks include chart question answering (ChartQA) [15, 16, 18, 32], chart captioning [6, 22], and automatic natural language annotation [19]. These tasks aim to convert chart content into human-readable forms to facilitate interpretation.

Recently, MLLMs have been applied to visualization understanding. For example, mChartQA [32] transforms charts into data tables to enable precise chart reasoning, while systems such as TinyChart [37] and ChartX [34] directly process chart images through vision-language models. Despite promising results, visualizations often involve spatially structured elements, such as axes, tick labels, and marks, which are not well represented in general pretraining.

[*]Equal contribution. E-mail: can.liu@ntu.edu.sg
[†]Equal contribution. E-mail: dachunlin@bytedance.com
[‡]E-mail: xiaoxiao.long@nju.edu.cn
[§]E-mail: yuxiao@fondant.design
[¶]E-mail: yu.zhang@cs.ox.ac.uk
[‖]Corresponding author. E-mail: yong-wang@ntu.edu.sg

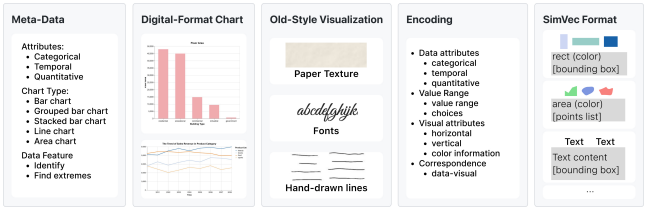

Figure 1: We convert the metadata into digital-format charts and then into historical-style charts. We also generate corresponding SimVec representations for the charts.

Moreover, current MLLMs for visualization tend to produce direct outputs without explicit reasoning, unlike human users, who typically engage in step-by-step inference when interpreting visualizations. To address these gaps, we propose SimVecVis, which introduces a compact vectorized representation of charts and incorporates chain-of-thought (CoT) reasoning into chart question answering. We further fine-tune our framework on models such as MiniCPM [14], which demonstrate good spatial awareness and support high-resolution visual input.

2.2 Visualization Datasets

Visualization researchers have constructed a series of visualization datasets [21] in the past few years. For example, VisImages [8] compiles visualizations extracted from diverse media. D3 search [11] crawled specific websites to collect visualizations created with D3 [5], while VizML [12] retrieves data and visual encoding specifications from an online gallery. VizNet [13] offers a large-scale corpus of 31 million data points gathered from open data repositories and online galleries. The reverse engineering visualization dataset [29] aggregates images from Vega Charts, news sites, and academic papers. Fu et al. [10] use dimension reduction technique to derive vector representations from infographic images. MASSVIS [4] automatically collects visualizations across fields from online websites. OldVisOnline and ZuantuSet gather historical visualizations before the computer era [26,39]. However, existing datasets overlook reasoning processes such as CoT and lack compact vector representations for reconstruction, which are the focus of SimVecVis.

3 SimVec Format

Vector formats (e.g., SVG) are widely used to represent visualizations. However, their structural flexibility and stylistic richness can hinder machine understanding. First, SVG supports nested <g> groups with transform attributes, leading to structural variability—semantically identical layouts may differ significantly in representation. Second, SVG provides multiple encoding methods for the same visual element. For example, a simple bar can be represented using a `rect`, `path`, `polygon`, or `points` element. Although these alternatives render identically to the human eye, they introduce syntactic inconsistency that complicates automated parsing and reasoning. Moreover, SVG files often include stylistic metadata, such as font-family, filters, or shadows, that are not essential for understanding the underlying data. To address these issues, we propose **SimVec**, a simplified vector format designed to retain the essential visual structure while enforcing a consistent, machine-readable representation. SimVec reduces complexity by: (1) flattening nested elements into an ordered list, (2) standardizing coordinates and color encodings, and (3) removing redundant styling. To support a variety of visualizations (**C1**), SimVec consists of four element types, as depicted in Table 1. SimVec is compact, reducing the token count by about 90% compared to the original SVG chart (e.g., generated using Vega-Lite). The tokenized length of the SVG may exceed the context window limits of many MLLMs, making it difficult for models to process.

Element	Description	Format	Example
Text	Represents textual content with position and styling information. Used for titles, labels, and annotations	`{text: content, bbox: [left, top, width, height], color: (h, s, l)}`	`{text "Title" [100, 50, 200, 30] hsl (0, 0, 18)}`
Rectangle	Used for bars, backgrounds, and other rectangular shapes. Defined by bounding box coordinates	`{rect: bbox[left, top, width, height], color: (h, s, l)}`	`{rect [100, 100, 50, 150] hsl (10, 15, 12)}`
Line	Represents axes, grid lines, and connecting lines. Defined by a series of points	`{line: points[(x1, y1), (x2, y2), ...], color: (h, s, l)}`	`{line [(0, 0), (100, 100)] hsl (0, 0, 5)}`
Polygon	Used for complex shapes and areas. Defined by a series of connected points forming a closed shape	`{polygon: points[(x1, y1), (x2, y2), ...], color: (h, s, l)}`	`{polygon [(0, 0), (50, 50), (100, 0)] hsl (5, 10, 15)}`

Table 1: All the coordinates and size mentioned above are described using a uniform value where the size is set to 1000. The color is represented in HSL color space and uniformized to [0-20] range.

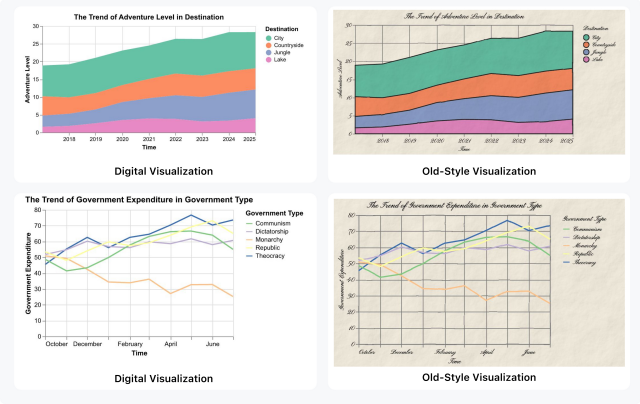

Figure 2: Historical-style visualizations with paper-texture, hand-drawn fonts, and hand-drawn lines.

4 Dataset Construction

To enable models to effectively understand charts, there are four design considerations for the SimVecVis dataset:

C1: Diverse Visualization Types. The dataset should include diverse visualization types and data attributes. Specifically, the dataset should contain common visualization types [3] such as bar charts, line charts, and area charts, covering different attributes like numerical, categorical, and temporal attributes from diverse topics.

C2: Accurate Data Features. We focus on data-centric QA tasks, specifically retrieving values and finding extremes [1]. Unlike trend detection, correlation estimation, or outlier detection, which can often be inferred from the overall shape of the chart, data QA requires precise decoding of the underlying quantitative values.

C3: Intermediate Reasoning. Many data QA questions require step-by-step reasoning—for instance, interpreting axis scales before mapping visual elements to values [28]. SimVecVis supports this process through chain-of-thought (CoT) annotations.

C4: Robustness to Imperfect Visual Inputs. To enhance practical relevance, SimVecVis incorporates charts from realistic settings, including hand-drawn visualizations sourced from historical documents. These visualizations often lack original data and feature noisy, irregular layouts, presenting unique challenges for perception and reasoning. Incorporating such visualizations can benefit the development of models capable of handling a wide range of visualizations beyond clean synthetic charts.

4.1 SimVecVis Dimensions

Fig. 1 illustrates the overall SimVecVis construction pipeline. Each instance contains the following components: a visualization image, its corresponding SimVec, and a set of question-answer (QA) pairs. Each QA pair is accompanied by a CoT description.

Visualization Bitmap Image. To ensure diversity, we prompt GPT-4o to generate meaningful data attributes based on commonly discussed topics. For example, in the energy domain, the LLM generates a categorical attribute (e.g., energy source), a temporal attribute (e.g., year), and corresponding quantitative values. We then populate these attributes with randomly synthesized data. The resulting datasets are then visualized using predefined templates for bar, line, or area charts. Color schemes are randomly assigned.

SimVec. The SVG format is then converted into a SimVec format, which captures a structured and vectorized representation of the visualization content. Ideally, if the elements of a visualization image can be reconstructed using SimVec, it will enable a precise extraction of axes and element position, size, and color. Therefore, our dataset includes SimVec for reconstructing the visualization structure and supporting further QA tasks (**C2**).

QA-pairs with CoT Descriptions. Given the metadata associated with each chart, we generate question-answer (QA) pairs. Among the low-level tasks [1], we focus on data tasks (**C2**), such as retrieving values and finding extremes. A question can be: "What is the proportion of gas in 2020?", with the corresponding answer being "35%". Each QA pair has a CoT description. When humans extract precise information from visualizations, they engage in an intermediate reasoning process (e.g., identifying axis mapping). Unlike previous datasets [32], our dataset incorporates reasoning steps in the question-answering process (**C3**), implementing what is known as the CoT approach [31]. To support step-by-step reasoning, we utilize axis metadata to guide the CoT process. For bar charts, this involves identifying the target bar and mapping its height to the corresponding value using the axis scale. A typical CoT trace includes the axis scale information, intermediate calculations, and the final answer. An example of a CoT description is *"For the Y-axis of the chart maps from 50 pixels to 450 pixels, corresponding to a percentage range of 0% to 100%. The height of the bar representing Gas in 2020 is 140 pixels. Thus, Gas in 2020 accounts for* $(140/(450-50)) \times 100 = 35\%$*."*

4.2 Preliminary Attempt on Historical Visualizations

While existing datasets focus on digitally rendered charts [17, 32] that are created using toolkits (e.g., Vega-Lite), historical visualizations present distinct challenges in analysis and interpretation. Vectorization and data extraction from historical visualizations [38] require heavy user involvement. As a preliminary step toward addressing these challenges, we mocked historical visualizations from digitally generated charts, aiming to expose models to stylistic variability beyond modern formats and improve their generalization and recognition abilities (**C4**). Fig. 2 compares digital visualizations with their corresponding historical-style representations. We employ the following steps to mock historical-style visualizations:

- **Paper Texture.** We simulate historical paper by incorporating natural textures and aging effects, including subtle surface irregularities and a slight yellow tint. These features enhance the visual distinction from modern, digitally rendered charts.
- **Hand-drawn Fonts.** We utilize specialized fonts that mimic handwriting styles seen in historical documents. These fonts provide a visually authentic handwritten appearance.
- **Hand-drawn Lines.** We reproduce the natural imperfections of manually drawn lines by introducing controlled variations in thickness and direction. These variations result in irregularities characteristic of hand-drawn charts.

SimVecVis comprises 2,999 visualizations, including 1,012 bar charts, 1,012 line charts, and 975 area charts. Each visualization is accompanied by a corresponding encoding output and a SimVec representation. The dataset includes 2,999 identification tasks and 5,642 extreme value detection tasks.

5 Experiments

To evaluate whether SimVecVis brings measurable improvements in visualization understanding, we conducted experiments centered on three key hypotheses, which examine: (1) the limitations of existing MLLMs, (2) the potential of CoT and SimVec to enhance visualization understanding, and (3) whether such gains are attributable to SimVec's ability to support visualization reconstruction.

- **H1:** Existing MLLMs cannot accurately support data understanding tasks without fine-tuning using SimVecVis.
- **H2:** CoT reasoning improves data QA accuracy, and is further enhanced with SimVec support.
- **H3:** Training MLLMs with SimVec representations improves their ability to reconstruct visualizations by providing a compact and structured encoding of visual elements.

Table 2: Accuracy is reported as the percentage of predictions with deviations of less than 5%, 10%, and 20% of ground truth values. Qwen-VL did not benefit from SimVec, likely due to its limited ability to accurately localize chart elements, which may have introduced additional computational burden during training.

Model	$< 5\%$	$< 10\%$	$< 20\%$
GPT-4o	16.54%	29.62%	42.69%
MiniCPM (zero-shot)	11.92%	17.69%	57.69%
DeepSeek-VL(zero-shot)	10.00%	17.31%	26.92%
Qwen-VL (zero-shot)	7.31%	13.46%	21.15%
MiniCPM (SimVec + QA w/ CoT)	**53.84%**	**69.23%**	**80.77%**
MiniCPM (QA w/ CoT)	29.23%	45.76%	69.23%
MiniCPM (QA w/o CoT)	26.92%	41.92%	25.38%
Qwen-VL (SimVec + QA w/ CoT)	5.38%	10.00%	18.08%
Qwen-VL (QA w/ CoT)	12.31%	21.54%	35.77%
Qwen-VL (QA w/o CoT)	11.54%	19.62%	31.15%

5.1 Performance Comparison of Zero-Shot Models

To validate **H1**, we assessed the zero-shot performance of several MLLMs. We selected the leading closed-source MLLM, GPT-4o, as well as several state-of-the-art open-source MLLMs: MiniCPM [14], DeepSeek-VL [23], and Qwen-VL [2]. In our experiment, data tasks rely on accurately localizing individual items. However, due to the lack of nominal references in scatter plots, pinpointing and directly identifying individual points is relatively challenging. Therefore, we selected bar charts, area charts, and line charts, as they allow for clearer localization of reference points through category and label information. These questions necessitate decoding the visual encoding in the chart to derive specific numerical values. The evaluation results are presented in Table 2. The evaluation metrics are presented as percentages, where accuracy is measured at three threshold levels: within 5%, 10%, and 20% deviation from the ground truth values. Among the four untrained models, GPT-4o performed best, which aligns with expectations for a high-capacity model. MiniCPM outperformed Qwen-VL and DeepSeek-VL, likely due to its specialized training in text localization, which enhanced its ability to directly extract numerical values.

H1 is supported. The overall performance of these MLLMs was moderate: they failed on tasks that required fundamental reasoning or simple calculations. In the following experiments, MiniCPM is chosen as the primary model, and Qwen-VL is used as a baseline.

5.2 Influence of CoT and SimVec on Data QA Accuracy

To validate **H2**, we compared three settings used for model training. We fine-tuned the model on 8 A100 40GB GPUs in about 14 hours (Taking MiniCPM for example). When addressing the same data question, the responses differ in the three settings: **(1) QA without CoT**: This setting takes a visualization image and a question as input and directly provides a numerical answer. **(2) QA with CoT**:

Table 3: Percentage of answers that have a difference rate under 5%, 10%, and 20%. The best performance is achieved by MiniCPM fine-tuned with chain-of-thought supervision and SimVec.

Model	Area Chart			Bar Chart			Line Chart		
	$< 5\%$	$< 10\%$	$< 20\%$	$< 5\%$	$< 10\%$	$< 20\%$	$< 5\%$	$< 10\%$	$< 20\%$
MiniCPM (SimVec + QA w/ CoT)	**38.16%**	**55.26%**	**69.74%**	**46.91%**	**65.43%**	**80.25%**	**70.87%**	**82.52%**	**89.32%**
MiniCPM (QA w/ CoT)	18.42%	39.47%	63.16%	33.33%	45.68%	74.07%	33.98%	50.49%	69.90%
MiniCPM (QA w/o CoT)	23.68%	36.84%	51.85%	20.99%	34.57%	51.85%	33.01%	51.46%	64.08%
MiniCPM (zero-shot)	2.63%	6.58%	14.47%	17.28%	22.22%	25.93%	14.56%	22.33%	33.01%

Compared to the direct answer setting, we add CoT description prior to the numerical answer. **(3) SimVec + QA with CoT**: In addition to the CoT, the model is additionally trained to predict the SimVec representation of the chart.

Table 2 shows that the model utilizing CoT supported by SimVec yields optimal performance. Training enhanced with CoT significantly outperforms the direct answer setting. For MiniCPM, the integration of SimVec information significantly enhanced its accuracy compared to using CoT reasoning alone. SimVec captures visual attributes of both text and marks, offering essential context (e.g., axes and encoding channels) required for effective CoT reasoning. However, for Qwen-VL, SimVec implementation did not yield improved accuracy, likely due to the model's limited ability to localize chart elements, with SimVec potentially introducing additional computational overhead during training. A detailed analysis of performance across various chart types for MiniCPM is illustrated in Table 3. Notably, MiniCPM (SimVec + CoT) surpasses all other model configurations in all chart types. The model performs substantially better on line charts compared to bar and area charts. This may be because line charts convey values more directly through position, whereas bar and area charts rely on height, which may involve stacking and thus introduce additional visual complexity. Despite improvements from SimVec and CoT, challenges remain. Current MLLMs often struggle to estimate spatial properties like bar height or line position, especially without explicit labels. Errors in early reasoning steps tend to propagate and affect final answers.

H2 is supported. CoT reasoning, especially when combined with SimVec, notably enhances MiniCPM's accuracy and achieves superior performance over current state-of-the-art MLLMs.

Table 4: Reconstruction Quality for Different Chart Types. The distance unit is 1/1000 of the image size.

Quality Metric	Line	Bar	Area
Text Hit Rate	99.79%	99.60%	99.83%
Text Similarity	98.37%	96.60%	98.72%
Text Center Distance	2.89	8.26	2.70
Element Color Distance	1.06	1.78	2.14
Element Position Distance	8.76	10.11	29.26

5.3 Reconstruction Capability using SimVec Format

To evaluate hypothesis **H3**, we assess the reconstruction capabilities using the SimVec format. We use MiniCPM (SimVec + CoT) to take an image as input and generate the corresponding SimVec as output. The dataset includes 100 images of bar, line, and area charts. Fig. 3 shows the original input image and the image rendered using the output reconstructed SimVec. The results demonstrate that different types of charts can be recovered to a satisfactory extent. As shown in Table 4, we calculate the hit rate and similarity as percentages. The distance is calculated in terms of the average number of pixels, where the image size is normalized to 1000 pixels. The quantitative tests include the text accuracy and graphics accuracy:

Text Accuracy Evaluation. To assess the model's text reconstruction capabilities, we employed multiple metrics: the *text hit rate* to measure the proportion of successfully recovered text elements, the *text similarity* using Levenshtein distance [36], and the *text center distance* to evaluate spatial accuracy of text placement. Experimental results demonstrate the model's robust performance, with text similarity reaching 98%. The average center distance deviation ranges from 0.27% to 0.83% of the image size (defined as the larger dimension of height or width), indicating high precision in spatial text recovery.

Figure 3: For each case (A, B, and C), the top panel displays the original input visualization, while the bottom panel shows the reconstructed result rendered using the SimVec output by the model.

Graphics Accuracy Evaluation. We employed the average pixel distance between predicted vertices and their corresponding ground truth as a metric to assess the reconstruction capabilities. The line elements in line charts demonstrated the highest precision with an average positional deviation of merely 0.88% of the image size. Bar elements showed comparable accuracy at 1% of image size, while area charts exhibited a slightly higher average distance of approximately 3%. These performance metrics align consistently with the data accuracy rankings presented in Table 3. For color fidelity assessment, we calculated the Euclidean distance between predicted and ground truth colors in the HSL color space (each dimension normalized to [0, 20]). The color differences' effect on overall perception is minor; for example, the differences are sufficient to identify distinct colors but unlikely to impair understanding.

These results validate **H3**, confirming that the SimVec format effectively supports high-fidelity reconstruction of both textual and graphical elements. Nevertheless, errors at the value level persist, mainly due to error accumulation in multi-step reasoning.

6 Conclusion and Future Work

We introduce SimVecVis that pairs bitmap charts with their SimVec representations and includes data QA tasks with CoT description, enabling supervised training for visualization understanding. We aim to expand the dataset to include infographics to support broader visualization scenarios.

Acknowledgments

This project is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Proposal ID: T2EP20222-0049). Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

REFERENCES

[1] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proc. IEEE InfoVis. Symp.*, pp. 111–117, 2005. doi: 10.1109/INFVIS.2005.1532136

[2] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. 2023. doi: 10.48550/ARXIV.2308.12966

[3] L. Battle, P. Duan, Z. Miranda, D. Mukusheva, R. Chang, and M. Stonebraker. Beagle: Automated extraction and interpretation of visualizations from the web. In *Proc. ACM Conf. Hum. Factors Comput. Syst.*, pp. 1–8, 2018. doi: 10.1145/3173574.3174168

[4] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister. What makes a visualization memorable? *IEEE Trans. Vis. Comput. Graph.*, 19(12):2306–2315, 2013. doi: 10.1109/TVCG.2013.234

[5] M. Bostock, V. Ogievetsky, and J. Heer. D^3: Data-Driven Documents. *IEEE Trans. Vis. Comput. Graph.*, 17(12):2301–2309, 2011. doi: 10.1109/TVCG.2011.185

[6] C. Chen, R. Zhang, E. Koh, S. Kim, S. Cohen, and R. Rossi. Figure captioning with relation maps for reasoning. In *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, pp. 1537–1545, 2020. doi: 10.1109/WACV45572.2020.9093592

[7] K. Cox, R. E. Grinter, S. L. Hibino, L. J. Jagadeesan, and D. Mantilla. A multi-modal natural language interface to an information visualization environment. *Int. J. Speech Technol.*, 4(3-4):297–314, 2001. doi: 10.1023/A:1011368926479

[8] D. Deng, Y. Wu, X. Shu, J. Wu, S. Fu, W. Cui, and Y. Wu. VisImages: A fine-grained expert-annotated visualization dataset. *IEEE Trans. Vis. Comput. Graph.*, 29(7):3298–3311, 2023. doi: 10.1109/TVCG.2022.3155440

[9] E. Fast, B. Chen, J. Mendelsohn, J. Bassen, and M. S. Bernstein. Iris: A conversational agent for complex tasks. In *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2018. doi: 10.1145/3173574.3174047

[10] X. Fu, Y. Wang, H. Dong, W. Cui, and H. Zhang. Visualization assessment: A machine learning approach. In *Proc. IEEE Vis. Conf.*, pp. 126–130, 2019. doi: 10.1109/VISUAL.2019.8933570

[11] E. Hoque and M. Agrawala. Searching the visual style and structure of D3 visualizations. *IEEE Trans. Vis. Comput. Graph.*, 26(1):1236–1245, 2019. doi: 10.1109/TVCG.2019.2934431

[12] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo. VizML: A machine learning approach to visualization recommendation. In *Proc. ACM Conf. Hum. Factors Comput. Syst.*, pp. 1–12, 2019. doi: 10.1145/3290605.3300358

[13] K. Hu, S. Gaikwad, M. Hulsebos, M. A. Bakker, E. Zgraggen, C. Hidalgo, T. Kraska, G. Li, A. Satyanarayan, and Ç. Demiralp. VizNet: Towards a large-scale visualization learning and benchmarking repository. In *Proc. ACM Conf. Hum. Factors Comput. Syst.*, pp. 1–12, 2019. doi: 10.1145/3290605.3300892

[14] S. Hu, Y. Tu, et al. MiniCPM: Unveiling the potential of small language models with scalable training strategies. 2024. doi: 10.48550/ARXIV.2404.06395

[15] K. Kafle, B. L. Price, S. Cohen, and C. Kanan. DVQA: understanding data visualizations via question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5648–5656, 2018.

[16] S. E. Kahou, V. Michalski, A. Atkinson, Á. Kádár, A. Trischler, and Y. Bengio. FigureQA: An annotated figure dataset for visual reasoning. In *Workshop Proceedings of International Conference on Learning Representations*, 2018.

[17] S. E. Kahou, V. Michalski, A. Atkinson, Á. Kádár, A. Trischler, and Y. Bengio. FigureQA: An annotated figure dataset for visual reasoning. In *Proc. Workshop Track Int. Conf. Learn. Represent.*, 2018. doi: 10.48550/arXiv.1710.07300

[18] D. H. Kim, E. Hoque, and M. Agrawala. Answering questions about charts and generating visual explanations. In *Proc. ACM Conf. Hum. Factors Comput. Syst.*, pp. 1–13, 2020. doi: 10.1145/3313831.3376467

[19] C. Lai, Z. Lin, R. Jiang, Y. Han, C. Liu, and X. Yuan. Automatic annotation synchronizing with textual description for visualization. In *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2020. doi: 10.1145/3313831.3376443

[20] C. Liu, Y. Han, R. Jiang, and X. Yuan. ADVISor: Automatic visualization answer for natural-language question on tabular data. In *Proc. IEEE Pac. Vis. Symp.*, pp. 6–15, 2021. doi: 10.1109/PacificVis52677.2021.00010

[21] C. Liu, R. Jiang, S. Tan, J. Yu, C. Yang, H. Shao, and X. Yuan. Datasets of visualization for machine learning, 2024. doi: 10.48550/arXiv.2407.16351

[22] C. Liu, L. Xie, Y. Han, X. Yuan, et al. AutoCaption: An approach to generate natural language description from visualization automatically. In *Proc. IEEE Pac. Vis. Symp.*, pp. 191–195, 2020. doi: 10.1109/PacificVis48177.2020.1043

[23] H. Lu, W. Liu, B. Zhang, B. Wang, K. Dong, B. Liu, et al. DeepSeek-VL: Towards real-world vision-language understanding, 2024. doi: 10.48550/arXiv.2403.05525

[24] Y. Luo, X. Qin, N. Tang, and G. Li. DeepEye: Towards automatic data visualization. In *Proc. IEEE Int. Conf. Data Eng.*, pp. 101–112, 2018. doi: 10.1109/ICDE.2018.00019

[25] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Trans. Vis. Comp. Graph.*, 13(6):1137–1144, 2007. doi: 10.1109/TVCG.2007.70594

[26] X. Mei, Y. Zhang, C. Yang, R. Shi, and X. Yuan. ZuantuSet: A collection of historical chinese visualizations and illustrations. In *Proc. ACM Conf. Hum. Factors Comput. Syst.*, 2025.

[27] 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 Trans. Vis. Comp. Graph.*, 25(1):438–448, 2018.

[28] S. Pinker. A theory of graph comprehension. *Artificial Intelligence and the Future of Testing*, pp. 73–126, 1990.

[29] J. Poco and J. Heer. Reverse-engineering visualizations: Recovering visual encodings from chart images. *Comput. Graph. Forum.*, 36(3):353–363, 2017. doi: 10.1111/cgf.13193

[30] Y. Sun, J. Leigh, A. Johnson, and S. Lee. Articulate: A semi-automated model for translating natural language queries into meaningful visualizations. In *Proc. Int. Symp. Smart Graph.*, pp. 184–195, 2010. doi: 10.1007/978-3-642-13544-6_18

[31] J. Wei, X. Wang, et al. Chain-of-thought prompting elicits reasoning in large language models. *Proc. NeurIPS*, 35:24824–24837, 2022. doi: 10.5555/3600270.3602070

[32] J. Wei, N. Xu, G. Chang, Y. Luo, B. Yu, and R. Guo. mChartQA: A universal benchmark for multimodal chart question answer based on vision-language alignment and reasoning, 2024. doi: 10.48550/arXiv.2404.01548

[33] K. Wongsuphasawat, D. Moritz, A. Anand, J. D. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans. Vis. Comp. Graph.*, 22(1):649–658, 2016. doi: 10.1109/TVCG.2015.2467191

[34] R. Xia, B. Zhang, et al. ChartX & ChartVLM: A versatile benchmark and foundation model for complicated chart reasoning, 2024. doi: 10.48550/arXiv.2402.12185

[35] B. Yu and C. T. Silva. FlowSense: A natural language interface for visual data exploration within a dataflow system. *IEEE Trans. Vis. Comput. Graph.*, 26(1):1–11, 2020. doi: 10.1109/TVCG.2019.2934668

[36] L. Yujian and L. Bo. A normalized levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1091–1095, 2007.

[37] L. Zhang, A. Hu, et al. TinyChart: Efficient chart understanding with visual token merging and program-of-thoughts learning, 2024. doi: 10.48550/arXiv.2404.16635

[38] Y. Zhang, B. Coecke, and M. Chen. MI3: Machine-initiated intelligent interaction for interactive classification and data reconstruction. *ACM Transactions on Interactive Intelligent Systems*, 11(3–4), Aug. 2021.

[39] Y. Zhang, R. Jiang, L. Xie, Y. Zhao, C. Liu, T. Ding, S. Chen, and X. Yuan. OldVisOnline: Curating a dataset of historical visualizations. *IEEE Trans. Vis. Comput. Graph.*, 30(1):551–561, 2024. doi: 10.1109/TVCG.2023.3326908