# Is GPT-4V (ision) All You Need for Automating Academic Data Visualization? Exploring Vision-Language Models' Capability in Reproducing Academic Charts

**Zhehao Zhang, Weicheng Ma, Soroush Vosoughi**
Department of Computer Science, Dartmouth College
{zhehao.zhang.gr, weicheng.ma.gr, soroush.vosoughi}@dartmouth.edu

## Abstract

While effective data visualization is crucial to present complex information in academic research, its creation demands significant expertise in both data management and graphic design. We explore the potential of using Vision-Language Models (VLMs) in automating the creation of data visualizations by generating code templates from existing charts. As the first work to systematically investigate this task, we first introduce AcademiaChart, a dataset comprising 2525 high-resolution data visualization figures with captions from a variety of AI conferences, extracted directly from source codes. We then conduct large-scale experiments with six state-of-the-art (SOTA) VLMs, including both closed-source and open-source models. Our findings reveal that SOTA closed-source VLMs can indeed be helpful in reproducing charts. On the contrary, open-source ones are only effective at reproducing much simpler charts but struggle with more complex ones. Interestingly, the application of Chain-of-Thought (CoT) prompting significantly enhances the performance of the most advanced model, GPT-4-V, while it does not work as well for other models. These results underscore the potential of VLMs in data visualization while also highlighting critical areas that need improvement for broader application. The dataset is available at https://github.com/zzh-SJTU/AcademiaChart.

## 1 Introduction

In academic research, effective data visualization is crucial for presenting complex information concisely (Inastrilla, 2023). However, crafting high-quality visualizations necessitates significant time and expertise in both data management and graphic design. Researchers may also face challenges in locating relevant documentation for Python libraries like Matplotlib (Hunter, 2007) or Seaborn (Waskom, 2021), further impeding knowledge dissemination and collaborative efforts in academia.

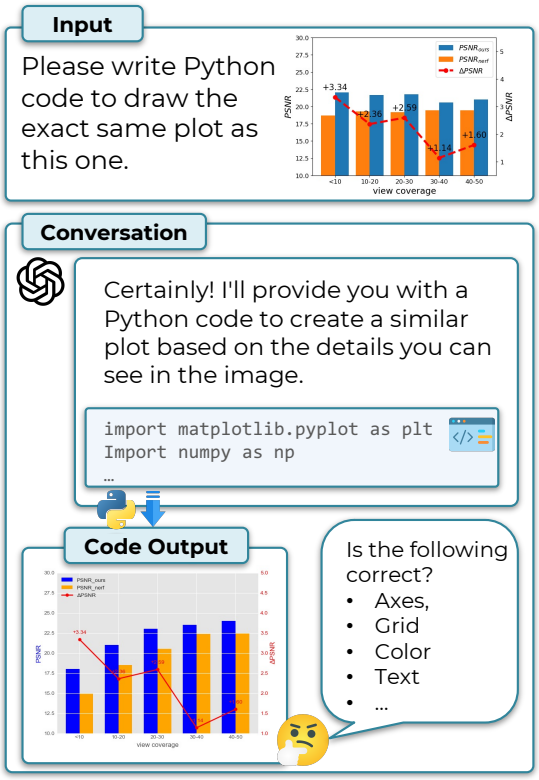



Figure 1: Our Task Formulation: We input instructional prompts from Vision-Language Models (VLMs) along with data visualization figures from top conferences, asking the models to write Python code that can reproduce the same figures.

A potential approach to tackle this issue is by exploring the feasibility of utilizing a model to generate a code template by providing an existing effective data visualization figure (*e.g.,* a screenshot), enabling researchers to replace it with their own data [1]. With the rapid development of Vision-Language Models (VLMs) (Radford et al., 2021; Li et al., 2022; Liu et al., 2023; Bai et al., 2023; Pichai, 2023; Wang et al., 2023), they have demonstrated

[1]Another two potential approaches are using an end-to-end image/text-to-image model for chart reproduction or employing natural language descriptions of charts as the input to LLMs; we discuss their feasibility in the Appendix A.1



*Findings of the Association for Computational Linguistics: EACL 2024*, pages 8271–8288

powerful performances in various vision-language tasks, including chartQA, chart-to-text, and chart-to-table (Masry et al., 2023). However, there are no prior works that systematically analyze VLMs' ability to reproduce charts or investigate whether they are helpful assistants for researchers in data visualization. To fill this gap, in this work, we focus on the task of chart-to-code generation, where the code is executed by an external code interpreter in an attempt to reproduce the original chart. This task is challenging because it requires the model to engage in two-fold multi-modality reasoning. As described in Figure 1, the model needs to first extract information from the given chart. Secondly, it must implicitly consider how the generated code will appear when executed by a code interpreter, ensuring that the resulting image matches the original chart effectively.

Although there is an existing dataset named SciCap (Hsu et al., 2021a) that contains charts from academic papers, it has the following limitations: (1) it **only** contains line graphs, which makes it less diverse, and (2) using optical character recognition (OCR) to extract figures from papers results in **low** resolution. To this end, we propose AcademiaChart, a dataset with 2525 real-life data visualization figures with captions from AI conferences. **(1)** Instead of using OCR, we directly extract figures from the source code of papers on Arxiv to get the original **high-resolution** charts with their captions. **(2)** Our dataset contains a diverse range of types of charts (as shown in Figure 3a) that can be used to conduct fine-grained analysis on VLMs. **(3)** Our data extraction pipeline (described in Figure 2) can be easily applied to collect a huge number of data without any human annotation.

We conduct comprehensive experiments with six state-of-the-art (SOTA) VLMs, including two closed-source models (GPT-4-V (OpenAI, 2023) and Gemini-Pro (Pichai, 2023)) and four open-source models, on our dataset. Both similarity-based metrics and fine-grained human evaluations indicate that there are still significant performance gaps between closed-source VLMs and open-source ones. For closed-source models, our results with comprehensive case studies indicate that they are indeed helpful in reproducing charts, although they can fail on some edge cases with extremely sophisticated structures. Additionally, Chain-of-thought (CoT) prompting can significantly improve the performance of the most powerful model, GPT-4-V. On the contrary, open-source models can only effectively reproduce charts with simple structures (e.g., line charts that contain a limited number of lines with default colors). CoT can even negatively affect the performance of these open-source VLMs. To our knowledge, this is the **first** systematic analysis of current VLMs' ability to reproduce charts and their helpfulness in assisting researchers in data visualization.

## 2 Related Works

### 2.1 VLMs and prompting

Large language models (LLMs) (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023; Zhang et al., 2022; OpenAI, 2023) have drawn significant improvements in NLP. Previous efforts in prompt engineering aimed to boost LLMs' capabilities, primarily using in-context learning by examples (Brown et al., 2020; Dong et al., 2022). Techniques like chain-of-thought (Wei et al., 2022; Kojima et al., 2022) and tree-of-thought (Yao et al., 2023) were later developed to enhance reasoning abilities in LLMs. Recently, Vision language models (VLMs) (Radford et al., 2021; Li et al., 2022; Alayrac et al., 2022; Li et al., 2023; Liu et al., 2023; Zhu et al., 2023; Bai et al., 2023; Wang et al., 2023; OpenAI, 2023) have become a prominent new research area. Currently, works (Yang et al., 2023a) on prompting VLMs are limited, mainly because most newly open-sourced models lack the capacity for such advanced capabilities (Wei et al., 2022). Most recently, Yang et al., 2023b conduct a qualitative study on GPT-4-V with case studies involving figure reproduction but did not include a comprehensive quantitative analysis. Han et al., 2023 introduce ChartLlama, a multi-modal model based on LLaVa-1.5 finetuned with chart-related data.

### 2.2 Chart datasets

There are recent datasets focusing on ChartQA (Masry et al., 2022; Kafle et al., 2018; Li et al., 2024; Obeid and Hoque, 2020) or Chart-To-Text generation (Masry et al., 2022; Kantharaj et al., 2022; Tang et al., 2023; Li and Tajbakhsh, 2023; Rahman et al., 2023; Zala et al., 2024) tasks where the source of the charts is from general domains. Wu et al., 2024 introduces a small dataset comprising 132 code-plot pairs from general domains. In scientific domains (Kahou et al., 2017; Karishma et al., 2023), Methani et al., 2020 introduce PlotQA, a large-scale dataset focusing on reasoning over scientific plots. However, these plots are generated

from templates rather than being derived from actual scientific studies and only focus on bar charts, line charts, and scatter plots. The most similar dataset, SciCap (Hsu et al., 2021a), uses OCR to extract charts with their captions from arXiv papers in PDF format. However, this dataset focuses **only** on line graphs, and the resolution of the figures is low. In contrast, our dataset contains over seven categories of charts with the original resolution.

## 3 Task Description

The task in this work is to leverage a VLM to interpret academic data visualizations and generate the corresponding code for their reproduction. The task can be succinctly outlined as follows: given a data visualization figure $F$ from an academic paper, accompanied by its caption $C$. An instructional prompt $P$ directs the model to generate code to replicate $F$. The VLM will perform the function $M(F, C, P)$, where: $M$ represents the model's processing capability. $F$ is the input figure. $C$ is the accompanying caption, providing context. $P$ is the user-provided prompt, specifically instructing the model to generate the code. The output will be a code snippet $S$, ideally in `Python`, utilizing common data visualization libraries (e.g., `Matplotlib` (Hunter, 2007), `Seaborn` (Waskom, 2021)). The task is formally expressed as:

$$S = M(F, C, P) \tag{1}$$

We visualize the task formulation in Figure 1.

## 4 AcademiaChart Dataset

This section provides an overview of the data collection process employed for extracting high-resolution charts and captions from academic papers. The motivation of AcademiaChart is to gather a diverse dataset from several prestigious conferences in the field of artificial intelligence (AI) including data mining, machine learning, computer vision, natural language processing, and robotics. We selected charts from the AI domain for our dataset because they can be generated by code, aligning with our goal to evaluate VLMs' image-to-code generation capabilities. The abundant availability of AI research, especially on arXiv, facilitates scaling our dataset. The data visualizations in AI research papers include diverse chart types and cover various AI subdomains, enabling broader applicability and generalization across fields

### 4.1 Data collection

The overall data collection pipeline is illustrated in Figure 2, Which is composed of the following two stages with 4 steps in total.

**Raw data scraping** In our research, we initially employ web scraping tools, such as BeautifulSoup, to extract relevant information (e.g., titles, authors, etc.) from the HTML content of publicly accessible sources, including the ACL Anthology, CVF Open Access, Github repositories, and the ArXiv homepage. Subsequently, we utilize the ArXiv API to obtain the ArXiv IDs associated with these papers, as determined from our scraped metadata. Using these IDs, we then download the source files, typically LaTeX projects, which comprise all text and original images from the papers. Finally, we apply regular expressions to detect patterns of image insertion in the LaTeX syntax. This enables us to locate the original images within the papers and extract their corresponding captions[2]. Contrary to the approach employed by Hsu et al., 2021b, which utilizes optical character recognition (OCR) for image extraction from PDF files, our method involves directly extracting images from source files. This ensures the retrieval of original images without any loss in resolution. Additionally, it enhances the accuracy of caption extraction, a significant improvement over the OCR-based method.

**Data filtering** As the raw figures from these academic papers are not all data visualization figures that can be reproduced using code, we employ the following steps to filter out the data visualization figures. We utilize a zero-shot prompting to input the figure into a VLM to determine if the figure is a data visualization figure. The prompt can be found in Table 3 in the Appendix.

Empirically, we find that current VLMs work well for correctly filtering out images that are not charts for data visualization, with LLaVa-V-1.5 (Liu et al., 2023) achieving over 95% accuracy and GPT-4-V surpassing 98% accuracy in a sample of 300 randomly selected examples[3].

### 4.2 Dataset analysis and statistics

Our comprehensive dataset includes 2525 image-caption pairs, carefully curated from a range

[2]We separately analyze the caption in Appendix B.

[3]Our dataset, re-filtered during human evaluation, excludes non-chart figures. The final dataset contains only charts, ensuring high quality despite minor filtering inaccuracies.

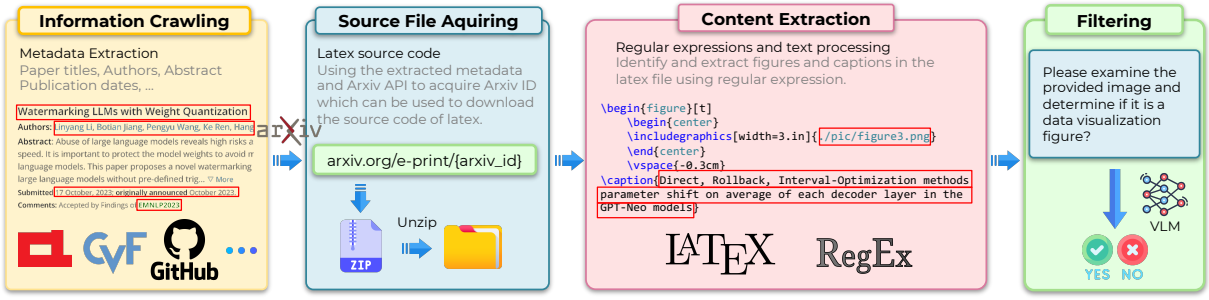
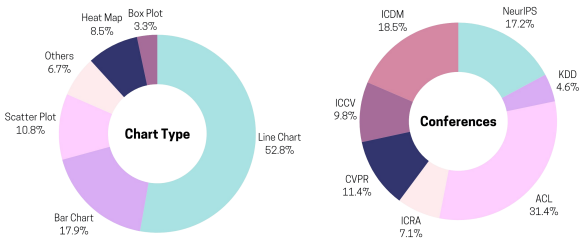



Figure 2: Our data collection pipeline. We first crawl various AI conference papers' arXiv information and then download their LaTeX source code. After that, we use regular expressions to extract patterns of figure insertion to locate the figure path and corresponding captions. Following this, we employ a VLM to filter out figures that are not data visualization. Our pipeline can be easily used to collect a large amount of high-resolution figures without any human annotations.

of prominent AI conferences including Association for Computational Linguistics (ACL), Conference on Computer Vision and Pattern Recognition (CVPR), International Conference on Computer Vision (ICCV), International Conference on Robotics and Automation (ICRA), Knowledge Discovery and Data Mining (KDD), International Conference on Data Mining (ICDM), and Neural Information Processing Systems (NeurIPS). The distribution of conferences that our charts are collected from is illustrated in Figure 3b.




(a) Chart type distribution (b) Data source distribution

Figure 3: The AcademiaChart dataset is collected from a wide range of AI conferences and encompasses a diverse range of chart categories.

Each data point is structured with the figure, the caption, the source file, and the data visualization type. Figure 3a illustrates the percentage of different categories of charts in AcademiaChart. The diversity of chart types in our dataset underscores the wide range of visualization techniques in AI research. Furthermore, the code used to generate these charts is generally not available online. Given that current VLM training requires (image, text) pairs, the likelihood of (chart, code) pairs from academic papers appearing online and being used is negligible, thereby minimizing concerns of data contamination.

## 5 Experiment

### 5.1 Models and baselines

We use the following VLMs in our experiments: (1) GPT-4-V (OpenAI, 2023), renowned for its superior performance across numerous vision-language tasks, thereby regarded as the most powerful VLM to date (Yang et al., 2023a; Taesiri et al., 2023; Cheng et al., 2023; Yue et al., 2023). (2) Gemini-Pro (Pichai, 2023), a recent release from Google, demonstrates performance comparable to GPT-4-V across various benchmarks (3) LLaVa-1.5-13b (Liu et al., 2023), an open-sourced VLM that features a linear projection mechanism that maps visual embeddings into the word embedding space of a LLM (4) MiniGPT-4 (Zhu et al., 2023), an open-source VLM which introduces a linear modality projection layer specifically designed to enhance visual comprehension capabilities. (5) Qwen-VL (Bai et al., 2023), an open-source VLM with a set of trainable query embeddings and a single-layer cross-attention module, facilitating the integration of image and text inputs. (6) CogVLM (Wang et al., 2023), a powerful open-source VLM combines image and text embeddings in its input space and integrates trainable visual layers into its textual transformer blocks for modality alignment. For all models, we use direct instructional prompting and zero-shot CCoT prompting (Kojima et al., 2022). The detailed prompt design and implementation details can be found in Appendix A.3 and A.2.

### 5.2 Evaluation protocol

As there is **no existing evaluation metric** to evaluate the quality of reproduced charts quantitatively, we use the following VLM as an evaluator, similarity-based metrics, and human evaluation.

**VLM as an evaluator:** [4] Inspired by recent findings that LLMs have the potential to evaluate NLP models (Chiang and Lee, 2023; Kamalloo et al., 2023), we use GPT-4-V and Gemini as evaluators to assess the quality of reproduced figures through an instructional prompt. As LLMs are known to generate unstable outputs, we employ GPT-4-V and Gemini only for comparative analysis rather than generating concrete scores. Specifically, we input the original figure along with two generated figures and ask GPT-4-V to determine which generated figure is more similar to the original. The complete prompt design is available in Appendix A.3. Given the limited resources and the large performance gap between closed-source and open-source VLMs, we restrict our comparative analysis to interactions between GPT-4-V and Gemini-Pro, and between their respective direct and CoT prompting on 100 randomly sampled data points.

**Automatic metric:** We use the cosine similarity between the embeddings of generated figures and labeled targets as a way to evaluate how well the VLM can reproduce the target chart. The embedding is extracted by a pre-trained Vision Transformer (ViT) (Dosovitskiy et al., 2020) model, which can capture semantics in images. This similarity, denoted as the Visual Embedding Similarity Score (VESS), is calculated as:

$$VESS(I, I') = \frac{E(I) \cdot E(I')}{\|E(I)\|\|E(I')\|} \quad (2)$$

where $E$ represents the process of obtaining embeddings, $I$ is the generated figure, and $I'$ is the target one. A higher value of $VESS$ indicates greater visual similarity, reflecting the model's accuracy in replicating target figures.

However, since ViT is not specifically pretrained on chart datasets, it may only capture some low-level features such as colors or lines, which do not align well with human preferences in chart creation. As a result, we propose the following human evaluation protocol as the **main** evaluation metric to validate VLMs' chart reproduction ability.

**Human evaluation** To get a deeper understanding of how current VLMs perform in different aspects of chart reproduction, we proposed a comprehensive human evaluation guideline that includes four main categories:

[4]We observe that GPT-4-V and Gemini align well with human preferences, ensuring the reliability of this evaluation.

**Structural Components Evaluation:** *Figure Type Accuracy:* Evaluators are asked to identify the type of chart (e.g., bar chart) and assess whether the reproduced chart correctly represents the original type. *Axis:* Rating to evaluate the presence, placement, scale, and accuracy of axis. *Tick Marks and Grid Lines:* Rating based on the correctness and placement of tick marks and grid lines. *Text Elements:* Evaluating the accuracy of text elements like titles, axis labels, legends, and annotations in terms of style and position.

**Stylistic Components Evaluation:** *Color Matching:* Assessing how closely the color palettes in the reproduced chart match the original. *Line/Bar/Marker Styles:* Rating the consistency and accuracy of line, bar, and marker styles compared to the original chart.

**Numerical Value Similarity:** Evaluators rate the visual accuracy of numerical values (e.g., bar heights) compared to the original chart. Since the actual raw data are not inputted into the charts, it poses a challenge for VLMs to estimate these values from the provided chart. However, this aspect is of lesser concern in practice, as researchers typically need to adapt the values to their own data.

**Practical Utility:** Rating the ease of adapting the reproduced chart for practical reuse, with or without modifications. This is the most important part of human evaluation, as it directly reveals the effectiveness of the VLMs in aiding researchers with chart reproduction.

All the above ratings are on a 5-point scale and accompanied along with detailed criteria. The complete human evaluation details and the interface can be found in Appendix B.1.

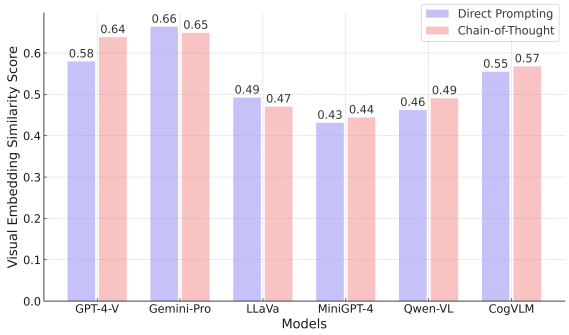



Figure 4: Comparison of the visual embedding similarity score (VESS) among different models: closed-source models (GPT-4-V and Gemini-Pro) outperform open-source ones. Chain of thought (CoT) yields no significant improvement except in GPT-4-V.

| Method | Type (ACC) | Axis | Marks&Grid | Text | Color | Style | Number | Utility |
|---|---|---|---|---|---|---|---|---|
| *Closed-source models* | | | | | | | | |
| GPT-4-V | 0.91 | 3.24 | 3.07 | 3.12 | 2.90 | 3.11 | 2.41 | 3.06 |
| GPT-4-V + CoT | 0.97 | 3.71 | 3.41 | 3.41 | 3.14 | 3.61 | 2.49 | 3.51 |
| Gemini-Pro | 0.87 | 3.68 | 3.08 | 2.98 | 3.11 | 3.07 | 2.33 | 2.93 |
| Gemini-Pro + CoT | 0.93 | 3.75 | 3.11 | 3.09 | 3.25 | 3.25 | 2.29 | 2.93 |
| *Open-source models* | | | | | | | | |
| LLaVa | 0.69 | 2.50 | 2.13 | 1.72 | 1.73 | 1.91 | 1.38 | 1.67 |
| LLaVa + CoT | 0.72 | 2.61 | 1.80 | 1.58 | 1.66 | 2.11 | 1.30 | 1.73 |
| MiniGPT-4 | 0.65 | 2.23 | 1.49 | 1.31 | 1.28 | 1.47 | 1.13 | 1.34 |
| MiniGPT-4 + CoT | 0.61 | 2.16 | 1.55 | 1.32 | 1.40 | 1.67 | 1.07 | 1.25 |
| Qwen-VL | 0.71 | 2.19 | 1.87 | 1.70 | 1.74 | 1.84 | 1.34 | 1.46 |
| Qwen-VL + CoT | 0.62 | 2.27 | 1.83 | 1.67 | 1.72 | 1.83 | 1.34 | 1.53 |
| CogVLM | 0.67 | 2.62 | 2.36 | 2.28 | 1.95 | 2.51 | 1.51 | 1.95 |
| CogVLM + CoT | 0.74 | 2.56 | 2.22 | 2.22 | 2.15 | 2.11 | 1.48 | 1.85 |

Table 1: Human evaluation results comparing the performance of various baselines from different perspectives. All our human evaluations utilize a 5-point Likert scale. GPT-4-V with CoT achieves the highest scores in the crucial criterion of practical utility, marking a significant improvement of 0.45 over direct prompting. GPT-4-V attains higher ratings than Gemini-Pro. A substantial gap is evident between closed-source models and open-source ones in these human evaluation results.

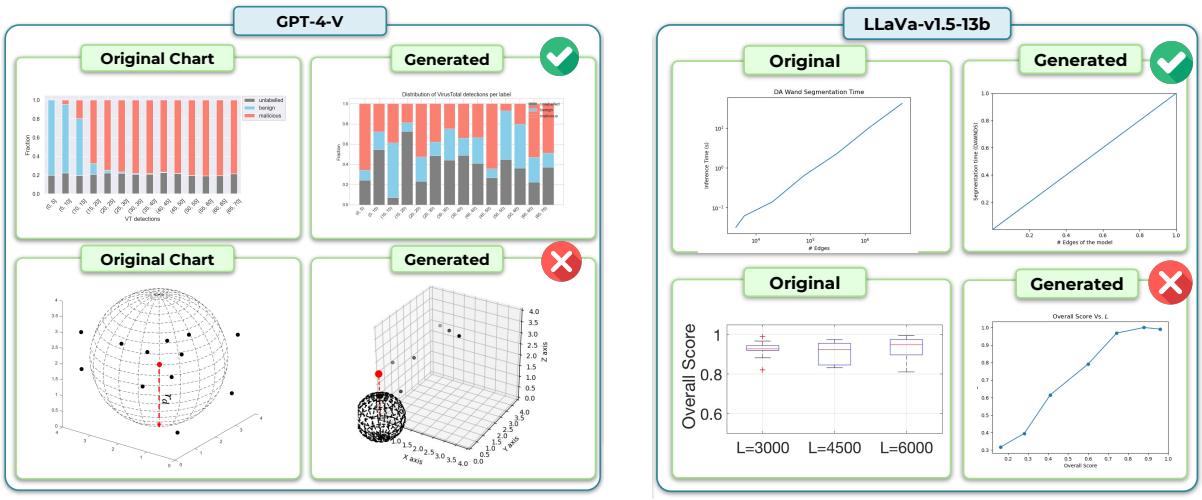
## 6 Results and Discussion

In this section, We analyze the experiment results by addressing the following 7 questions.


Figure 5: GPT-4-V's fine-grained human evaluation scores across various chart categories. The model exhibits consistent scoring patterns, performing particularly well in the areas of axis and tick marks. Notably, GPT-4-V achieves its highest scores in line and scatter chart analyses.

**RQ1: Is the current SOTA VLM capable enough to accurately reproduce charts in academic papers and assist researchers in data visualization?** Answer: **Partially yes**. As shown in Table 1, GPT-4-V with CoT Prompting achieves the best performances. Notably, it can accurately reproduce 97% of figure types, excelling in rendering tick marks, grid lines, and various line and bar types (solid, dashed, dotted, etc.), as well as marker styles. Additionally, as depicted in Figure 4, Gemini-Pro demonstrates the highest VESS rating. Table 1 also shows Gemini-Pro's superior performance in axis and color reconstruction. These findings suggest that the current SOTA VLMs, such as GPT-4-V and Gemini-Pro, can indeed alleviate the workload of researchers by providing efficient and usable code templates. However, Figure 6a shows a case in which GPT-4-V cannot reproduce a similar chart, even with CoT prompting. Empirically, we find that GPT-4-V may struggle more to reproduce charts that are not one of the most frequently used categories such as line charts or scatter plots.

**RQ2: Is the current SOTA VLM helpful in assisting researchers in data visualization?** Answer: **Yes**. As shown in Table 1, GPT-4-V with CoT prompting achieves a high utility score of 3.51, indicating that the code requires only minor adjustments for reuse with new data sets, making it highly practical. As the generated code sample shown in Figure 9 in the Appendix, the code that draws a data visualization chart using commonly used Python libraries typically consists of different stages, such as library loading, data loading, axes adjusting, style setting, and so on, which can be complex. However, with the assistance of a VLM to generate a well-designed code template, we only need to modify the data loading, which usually involves simply changing the data array. This simplification makes the process accessible even to non-experts. To validate our Figure-to-Code task setting, we sample 30 data points and simulate new data adaptations for two participant groups. One




(a) Positive and negative cases for GPT-4-V

(b) Positive and negative cases for LLaVa-v1.5

Figure 6: Case studies comparing a closed-source model (GPT-4-V) with an open-source model (LLaMa-v1.5). GPT-4-V can replicate similar styles across a wide range of charts but fails in some edge cases. Open-source models perform well only with charts having simple structures.

group used a GPT-4-V-generated code template, while the other created charts from scratch with online resources. The template group averaged 43.2 seconds to complete the task, significantly faster than the 246.4 seconds for the scratch group. Furthermore, the template group scored higher in practical utility (3.4 vs. 2.7). These results support the efficiency and quality improvements provided by using VLM-generated code templates.

**RQ3: Which one is better for chart reproduction, Gemini-Pro or GPT-4-V?** Answer: **GPT-4-V**. The human evaluation results in Table 1 indicate that GPT-4-V consistently outperforms other models in terms of practical utility and most of the other criteria. Besides, according to results using GPT-4-V as the evaluator, the model preferred 59% of the figures generated by GPT-4-V and 36% of the figures generated by Gemini-Pro. In 62% of cases, Gemini-Pro showed a preference for charts generated by GPT-4-V, while in 34% of cases, it preferred its own charts. These results indicate that no biases were observed where a VLM prefers charts generated by the same model. Although Figure 4 shows that Gemini-Pro outperforms GPT-4-V in VESS, indicating better low-level similarity between generated charts and the labels, we do not consider Gemini-Pro as a better model because alignment with human intent is more important.

**RQ4: How do open-source and closed-source VLMs compare in terms of chart reproduction capabilities?** Answer: Closed-source VLMs (e.g., GPT-4-V and Gemini-Pro) are **much better** than open-source models. As shown in Table 1 and Figure 4, it is evident that no open-source model with direct prompting or CoT can outperform GPT-4-V or Gemini-Pro in both human evaluation and VESS. Noticeably, in all fine-grained human evaluation criteria, closed-source models outperform the open-source models. Among these human evaluation criteria, the most important practical utility rating; the gap between the best-performed open-source model, CogVLM, and Gemini-Pro is still 0.98. As the cases shown in Figure 6b, we find that open-source VLMs can only effectively reproduce charts with simple structures and default styles [5]. Consequently, we conclude that SOTA open-source VLMs are far from practical to help assist researchers in data visualization.

**RQ5: In chart reproduction, what aspects do VLMs excel in, and in which aspects do they fall short?** Answer: They excel in reproducing **axis** but are poor at **numerical** approximation. As shown in Table 1, we observe that across all models, the highest human evaluation scores are achieved in axis reproduction. Intuitively, it is likely because charts with two axes, one on the left and one on the bottom, are the most common configuration in both the pre-training and test datasets. Empirically, we also find that VLMs struggle to accurately reproduce axis in less common configurations where

[5] More case studies comparing different models and examples of generated code can be found in Appendix B.2.

the axes are not positioned one on the left and one on the right. Notably, all models score lowest in human evaluations regarding numerical estimation. This suggests a struggle to accurately estimate the numerical values of bars or lines in original figures. However, this aspect is of lesser concern, as we typically replace these estimated values with actual data for practical use.

**RQ6: How does the performance of GPT-4-V vary across different types of charts?** In Figure 5, we present the fine-grained human evaluation scores of GPT-4-V across various types of charts. The figure reveals a consistent scoring pattern for GPT-4-V, with notably better performance in reproducing axis and tick marks, while its capability in numerical estimation is comparatively weaker. Besides, in terms of overall practical utility, GPT-4-V achieves its highest score with line charts and the lowest with bar charts, though the difference is not substantial. Consequently, we conclude that GPT-4-V possesses a balanced ability in reproducing different categories of charts in academic papers.

**RQ7: Is Chain-of-Thought (CoT) prompting still effective for VLMs in chart reproducing for most VLMs?** Answer: **No**, CoT only has noticeable effectiveness for GPT-4-V. As shown in table 1, CoT significantly improves human evaluation scores only for the GPT-4-V, with enhancements observed in 7 out of 8 criteria. Notably, CoT increases the most important practical utility score by a considerable margin of 0.45. Additionally, Figure 4 indicates that CoT also substantially improves the VESS by a notable margin of 0.059. To further investigate the effectiveness of CoT on GPT-4-V, we randomly sampled 200 data points and ask human annotators to select which one is more similar to the original chart (with CoT, without CoT, or equal). As shown in Figure 7, there are significantly more data points for which human annotators prefer the CoT approach. Besides, we use GPT-4-V as an evaluator to compare the figures generated by direct prompting and CoT prompting with the original ones. Results show that for 76% of data points, GPT-4-V prefers the figures generated by CoT prompting; for 14%, it prefers those from direct prompting; and for 10%, it shows no preference. These results suggest that CoT remains effective for the most powerful VLM, GPT-4-V. In contrast, CoT appears **not** to be effective for other, less powerful models. For Gemini-Pro, although CoT leads to improvements in 6 out of 8 criteria

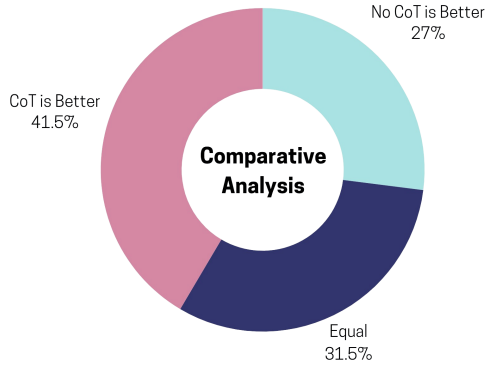



Figure 7: Comparative Analysis of GPT-4-V with and without CoT. The data shows a significantly higher number of instances where charts generated with CoT are preferred, indicating the effectiveness of CoT in enhancing GPT-4-V's performance in chart reproduction

and GPT-4-V as an evaluator prefers CoT generated figures in 53% of cases, the practical utility score remains the same as with direct prompting. Furthermore, Figure 4 shows that Gemini-Pro with CoT even gets a lower VESS score. For open-source VLMs, the performance difference between CoT and direct prompting is minimal, and CoT actually results in a lower practical utility score for 2 out of 4 open-source models. Previous works (Wei et al., 2022) have found that CoT reasoning is an emergent ability associated with increasing model scale in text-only LLMs. Our results suggest that for VLMs, CoT reasoning also seems to emerge only when the model's size and capability reach a certain threshold.

## 7 Conclusion

In this work, we explore the use of VLMs for automating the creation of data visualizations in academic research. We introduce AcademiaChart, a novel dataset comprising 2525 high-resolution figures from AI conferences, extracted directly from LaTeX source codes without using OCR. Subsequently, we conduct comprehensive experiments with six SOTA VLMs. Our findings reveal that GPT-4-V and Gemini-Pro are particularly effective in generating complex charts, underscoring their significant value in simplifying the data visualization process. Notably, GPT-4-V emerges as the best model for chart reproduction, being the only one where the utility of CoT prompting shows significant improvement. In contrast, open-source models are limited to reproducing much simpler charts and do not demonstrate the emergence of CoT capability. We believe the future trajectory of automated

data visualization in academia is promising, with VLMs expected to play a fundamental role in its advancement.

## Limitations

Our work has several limitations :

1. **Textual Prompt Focus:** Our methodology primarily relied on textual prompts, without exploring the potential enhancements that visual prompt engineering could offer in the performance of VLMs.

2. **Rapid Evolution of VLMs:** The field of VLMs is evolving rapidly. We utilized the SOTA VLMs as of December 2023. However, advancements post-2023 might offer different insights or improved capabilities not captured in this study.

3. **Dataset Diversity and Scope:** Our dataset, AcademiaChart, comprises figures from AI conferences only. This limitation restricts the diversity and might omit complex visualizations that are not code-based, affecting the generalizability of our findings.

4. **Dataset Size:** AcademiaChart is a test-only dataset with a relatively small number of data points. This smaller scale may impact the applicability of our findings across more diverse and extensive datasets.

5. **Limited Accurate Quantitative Evaluation Metric:** As described in Section 5.2, no existing metric adequately focuses on our task. Determining how to accurately and objectively compare a generated data visualization to a gold standard remains an open problem that requires future resolution. Although we have employed comprehensive embedding similarity-based methods and human evaluation, developing a robust metric is crucial for advancing research in this area.

In summary, while our study provides valuable insights into the use of VLMs for data visualization in academic research, these limitations highlight areas for future research and development.

## References


Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. 2023. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2:3.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sijie Cheng, Zhicheng Guo, Jingwen Wu, Kechen Fang, Peng Li, Huaping Liu, and Yang Liu. 2023. Can vision-language models think from a first-person perspective? *arXiv preprint arXiv:2311.15596*.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. 2023. Chartllama: A multimodal llm for chart understanding and generation.

Ting-Yao Hsu, C Lee Giles, and Ting-Hao Huang. 2021a. SciCap: Generating captions for scientific figures. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3258–3264, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ting-Yao Hsu, C Lee Giles, and Ting-Hao'Kenneth' Huang. 2021b. Scicap: Generating captions for scientific figures. *arXiv preprint arXiv:2110.11624*.

John D Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(03):90–95.

Carlos Rafael Araujo Inastrilla. 2023. Data visualization in the information society. In *Seminars in Medical Writing and Education*, volume 2, pages 25–25.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. 2017. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.

Shankar Kantharaj, Rixie Tiffany Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022. Chart-to-text: A large-scale benchmark for chart summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4005–4023, Dublin, Ireland. Association for Computational Linguistics.

Zeba Karishma, Shaurya Rohatgi, Kavya Shrinivas Puranik, Jian Wu, and C. Lee Giles. 2023. Acl-fig: A dataset for scientific figure classification.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. 2024. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models.

Shengzhi Li and Nima Tajbakhsh. 2023. Scigraphqa: A large-scale synthetic multi-turn question-answering dataset for scientific graphs. *arXiv preprint arXiv:2308.03349*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *NeurIPS*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.

Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. 2023. UniChart: A universal vision-language pretrained model for chart comprehension and reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14662–14684, Singapore. Association for Computational Linguistics.

Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1527–1536.

Jason Obeid and Enamul Hoque. 2020. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 138–147, Dublin, Ireland. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Sundar Pichai. 2023. Introducing gemini: Google's most capable ai model yet. Google Blog.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Raian Rahman, Rizvi Hasan, Abdullah Al Farhad, Md Tahmid Rahman Laskar, Md Hamjajul Ashmafee, and Abu Raihan Mostofa Kamal. 2023. Chartsumm: A comprehensive benchmark for automatic chart summarization of long and short summaries. *arXiv preprint arXiv:2304.13620*.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Mohammad Reza Taesiri, Tianjun Feng, Cor-Paul Bezemer, and Anh Nguyen. 2023. Glitchbench: Can large multimodal models detect video game glitches? *arXiv preprint arXiv:2312.05291*.

Benny J Tang, Angie Boggust, and Arvind Satyanarayan. 2023. Vistext: A benchmark for semantically rich chart captioning. *arXiv preprint arXiv:2307.05356*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. 2023. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*.

Michael L. Waskom. 2021. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Chengyue Wu, Yixiao Ge, Qiushan Guo, Jiahao Wang, Zhixuan Liang, Zeyu Lu, Ying Shan, and Ping Luo. 2024. Plot2code: A comprehensive benchmark for evaluating multi-modal large language models in code generation from scientific plots. *arXiv preprint arXiv:2405.07990*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023a. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023b. The dawn of lmms: Preliminary explorations with gpt-4v(ision).

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 2023. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*.

Abhay Zala, Han Lin, Jaemin Cho, and Mohit Bansal. 2024. Diagrammergpt: Generating open-domain, open-platform diagrams via llm planning. In *COLM*.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

# A Appendix

## A.1 Preliminary Studies

Recent advances in generative AI have enabled end-to-end text-to-image generative models (Ramesh et al., 2022; Betker et al., 2023; Zhang et al., 2023) to realize significant enhancements in performance. With the introduction of image/text-to-image diffusion models (Rombach et al., 2022), it's now possible to generate visually impressive images simply by entering a text description. As a result, we conducted the following case studies for our task using DALL·E-3 (Betker et al., 2023), which is one of the SOTA image/text-to-image models integrated into the ChatGPT web interface.

As shown in the Figure 8, although DALL·E-3 is widely recognized for its superior performance in creative image generation, we observe that it struggles to accurately reproduce charts from academic papers. In particular, it performs poorly in generating text that matches the reference of the original figure. Additionally, the colors in the generated image are often completely different from those in the original. While some components of the generated figure bear a resemblance to the original, the overall quality falls short of the standards required for a chart in academic papers. We also implemented the self-refinement strategy as described in (Madaan et al., 2023) and attempted to have the model regenerate the image; however, the results remained unsatisfactory.

Overall, we conclude that utilizing current end-to-end image/text-to-image generation models is

Prompt Design for Baselines

| | |
|---|---|
| Zero-shot | |
| Image | figure_image |
| Instruction: | This is a data visualization figure from an academic paper with the caption of figure_caption Please write Python code using matplotlib to draw the exact same plot as this one and save it as a PNG file with 300dpi. |
| Response: | |
| Zero-shot-CoT | |
| Image | figure_image |
| Instruction: | This is a data visualization figure from an academic paper with the caption of figure_caption Please write Python code using matplotlib to draw the exact same plot as this one and save it as a PNG file with 300dpi. Let's think step-by-step |
| Response: | |

Table 2: Prompt designs of all prompting baselines in the experiment

Prompt Design for Data Filtering

| | |
|---|---|
| Data Filtering | |
| Image | original_figure_image |
| Instruction: | Please examine the provided image and determine if it is a Bar Chart, Line Chart, Pie Chart, Histogram, Scatter Plot, Box Plot, etc., to present experiment results |
| Response: | |

Table 3: Prompt designs of data filtering

| | |
|---|---|
| GPT-4-V-Eval | |
| Original Image | original_figure_image |
| Generated Image 1 | generated_figure_image |
| Generated Image 2 | generated_figure_image |
| Instruction: | There are three data visualization figures: the first one is the gold standard, and the other two are figures generated by models. Please compare the first figure with the others from perspectives such as axes, tick marks, grid lines, text elements, color palettes, line types, bar types (solid, dashed, dotted, etc.), and marker styles to determine which of the remaining two is more similar to the first one: A. The second one is more similar. B. The third one is more similar. C. Equal. |
| Response: | |

Table 4: Prompt designs of VLM as an evaluator

not a practical approach for automating academic data visualization. Consequently, our setting, which employs VLMs to first generate code and then execute this code to produce the figure, currently stands as the most practical method for this task.

Another potential approach to utilizing LLMs for automating data visualization is to use natural language descriptions as input, instead of the figures themselves. We opt for image input in our setting because it allows us to simply take a screenshot of a well-designed data visualization figure. This method is not only easy and straightforward but also captures all the detailed information that could potentially be useful for the model to reproduce the figure. In contrast, verbally describing every detail of a chart—including textual content, colors, stylistic elements, positions, and axis structures—is nearly impossible. Additionally, in the data collection process, it is much harder to obtain high-quality description-figure pair data than to just collect the figures.

### A.2 Implementation details

For our experiments, we used OpenAI's `gpt-4-vision-preview` API for GPT-4-V, Google's `gemini-pro-vision` API for Gemini, the `llava-v1.5-13b` checkpoint for LLava, the `llama2`-incorporated version for Mini-GPT-4, the `cogvlm-chat-hf` checkpoint with 16-bit quantization for CogVLM, and `Qwen-VL-Chat` checkpoint for Qwen-VL. We set all models' temperatures to 0.1 for reproducibility. As human evaluation and API calls are extremely expensive, we randomly sample 437 data points for all our experiments. Empirically, we find the generated code is executable in most cases. In cases where execution fails, we input the error message into the VLM to revise the code, ensuring executability. For postprocessing, we add a line to save the image to a specific local path.

### A.3 Prompt Design

The complete prompt design for the main experiments can be found in Table 2. As stated before, we employ the zero-shot setting because a large

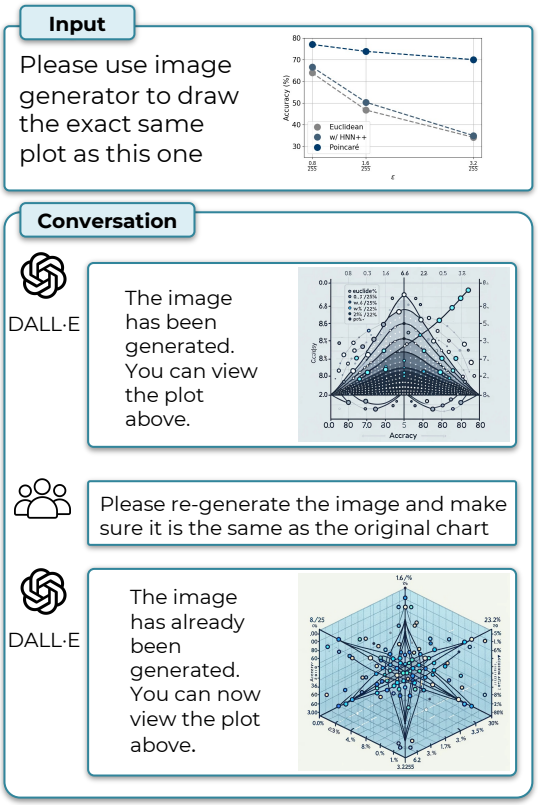



Figure 8: An example of using an end-to-end text-to-image model, DALL·E 3, to perform our task is provided. The significant difference between the generated figure and the original one indicates that this type of model is not practical for our task

```
import matplotlib.pyplot as plt
import numpy as np
categories = ['(0, 5)', '(5, 10)', ...']
n_categories = len(categories)
malicious = np.random.rand(n_categories)
benign = np.random.rand(n_categories)
unlabelled = np.random.rand(n_categories)
total = malicious + benign + unlabelled
malicious_fraction = malicious / total
benign_fraction = benign / total
unlabelled_fraction = unlabelled / total
x = np.arange(n_categories)
plt.figure(figsize=(10, 5))
plt.bar(x, unlabelled_fraction, color='gray’,
          label='unlabelled')
plt.bar(x, benign_fraction,
          bottom=unlabelled_fraction,
          color='skyblue', label='benign')
plt.bar(x, malicious_fraction, bottom= unlabelled_fraction +
          benign_fraction, color='salmon’,
          label='malicious')
plt.xticks(x, categories, rotation=45, ha="right")
plt.ylabel('Fraction')
plt.title('Distribution of VirusTotal detections per label')
plt.legend()
plt.show()
```

Figure 9: A piece of code generated by GPT-4-V given the image input of Figure 6a.

number of VLMs do not support multi-image input. The prompt design for GPT-4-V as an evaluator can be found in Table 4

# B Ablation Study

To separately analyze the effectiveness of captions in VLMs' performances on chart reproduction, we conduct an ablation study on 50 randomly sampled data points using (1) only the chart image and (2) only the caption with GPT-4-V and Gemini-Pro.

As shown in Table 5, using only the image without the caption as input can yield results similar to combining the image and caption, with only the text part showing a slight decrease. Conversely, using only the caption results in much worse performance in reproducing the figures, as expected. Consequently, the effect of adding the caption is not significantly effective in aiding chart reproduction.

## B.1 Human Evaluation Details and Interface

We include three graduate students majoring in computer science, each possessing sufficient experience in data visualization, to participate in the human evaluation. We randomly select 200 data points for overlap and compute the inter-annotator agreement. The average inter-annotator agreement score of 0.91 indicates the stability of the human evaluation. The complete interface for human evaluation is illustrated in Figures 16 and 17.

## B.2 More Case Studies

In this section, we provide additional case studies illustrated in Figures 10, 11, 12, 13, 14, and 15. We observe a significant performance disparity between open-source VLMs and advanced models like GPT-4-V or Gemini-Pro. The open-source models primarily generate basic curves, such as sine or cosine waves or straight lines, often overlooking the intricate details of the original figures, such as markers. In contrast, GPT-4-V or Gemini-Pro can reproduce charts more accurately, making their code templates particularly useful for researchers in data visualization. An example piece of code generated by GPT-4-V can be found in Figure 9.

| Model (Method) | Type (ACC) | Axis | Marks & Grid | Text | Color | Style | Number | Utility |
|---|---|---|---|---|---|---|---|---|
| GPT-4-V (only image) | 0.90 | 3.2 | 3.1 | 3.0 | 2.9 | 3.1 | 2.3 | 3.0 |
| Gemini-Pro (only image) | 0.86 | 3.6 | 3.0 | 2.7 | 3.0 | 3.1 | 2.3 | 2.9 |
| GPT-4-V (only caption) | 0.30 | 1.6 | 1.2 | 1.1 | 1.1 | 1.0 | 1.0 | 1.1 |
| Gemini-Pro (only caption) | 0.28 | 1.5 | 1.1 | 1.1 | 1.1 | 1.0 | 1.0 | 1.1 |

Table 5: Human evaluation results comparing performance using only image versus only caption with GPT-4-V and Gemini-Pro

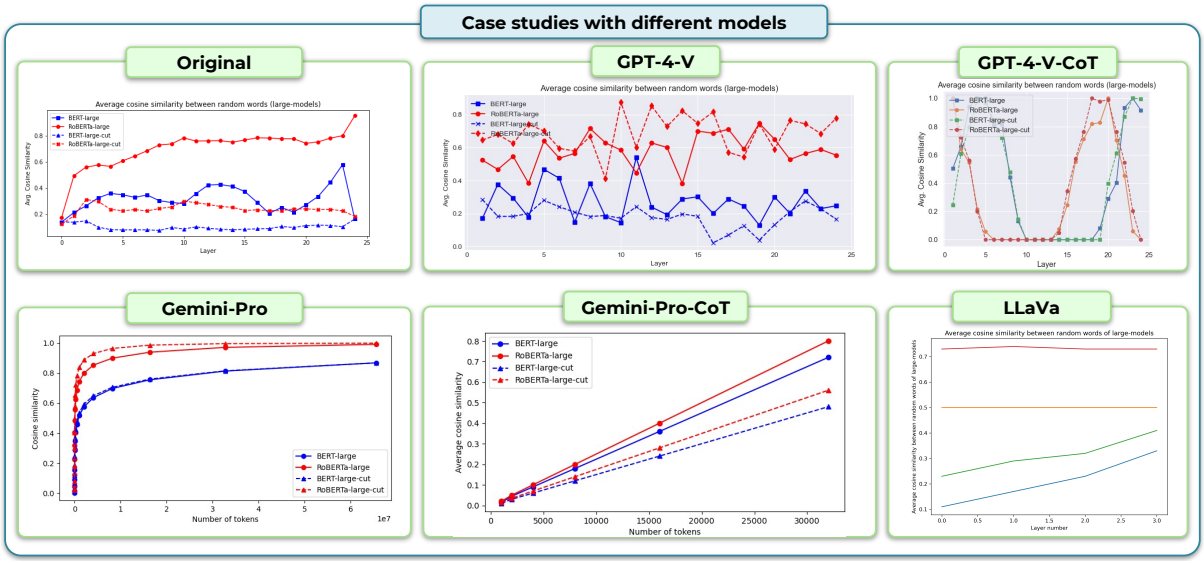



Figure 10: More case studies

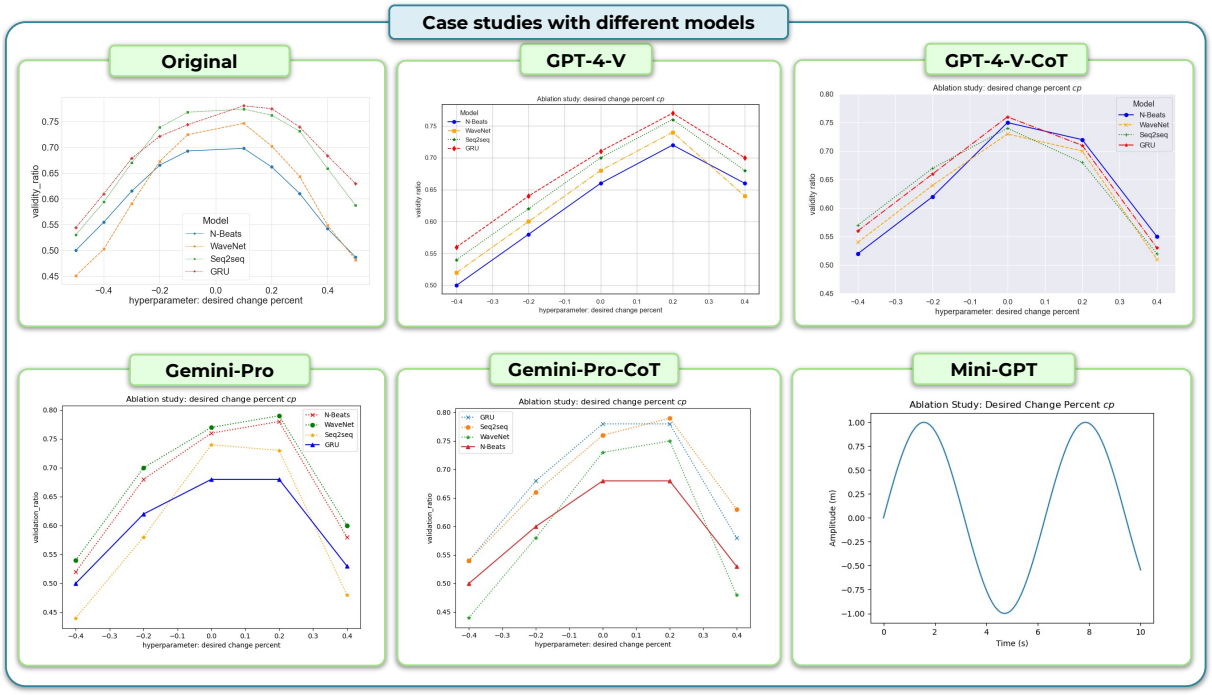



Figure 11: More case studies

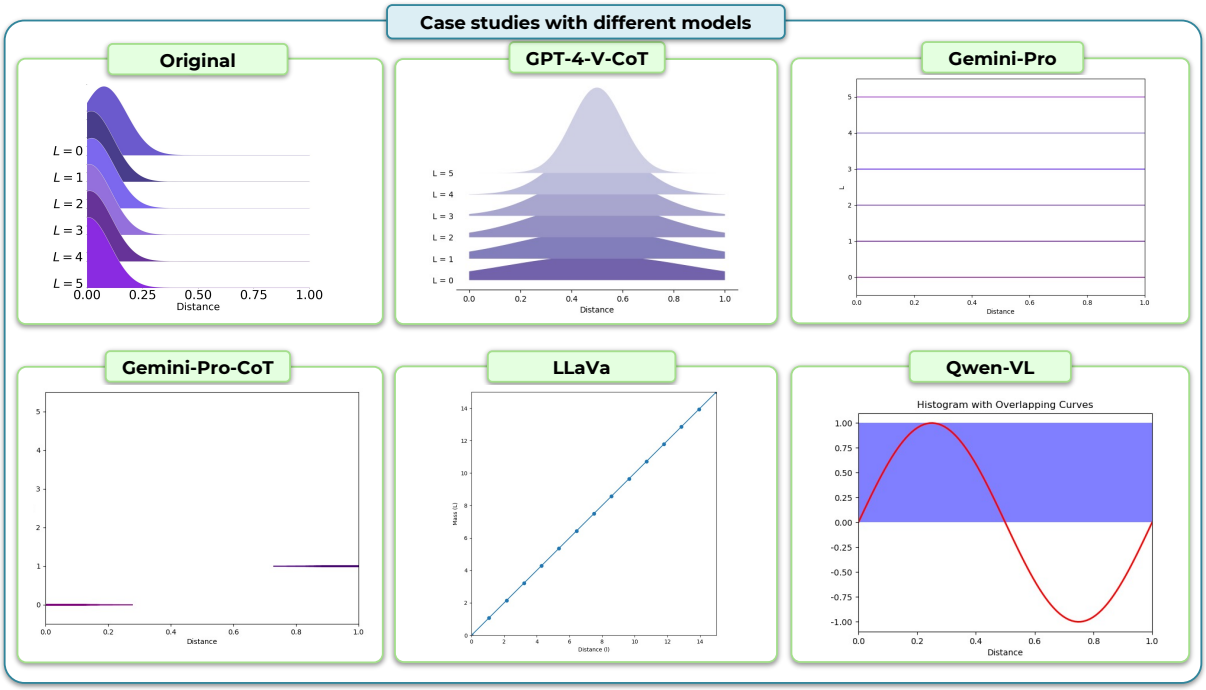



Figure 12: More case studies

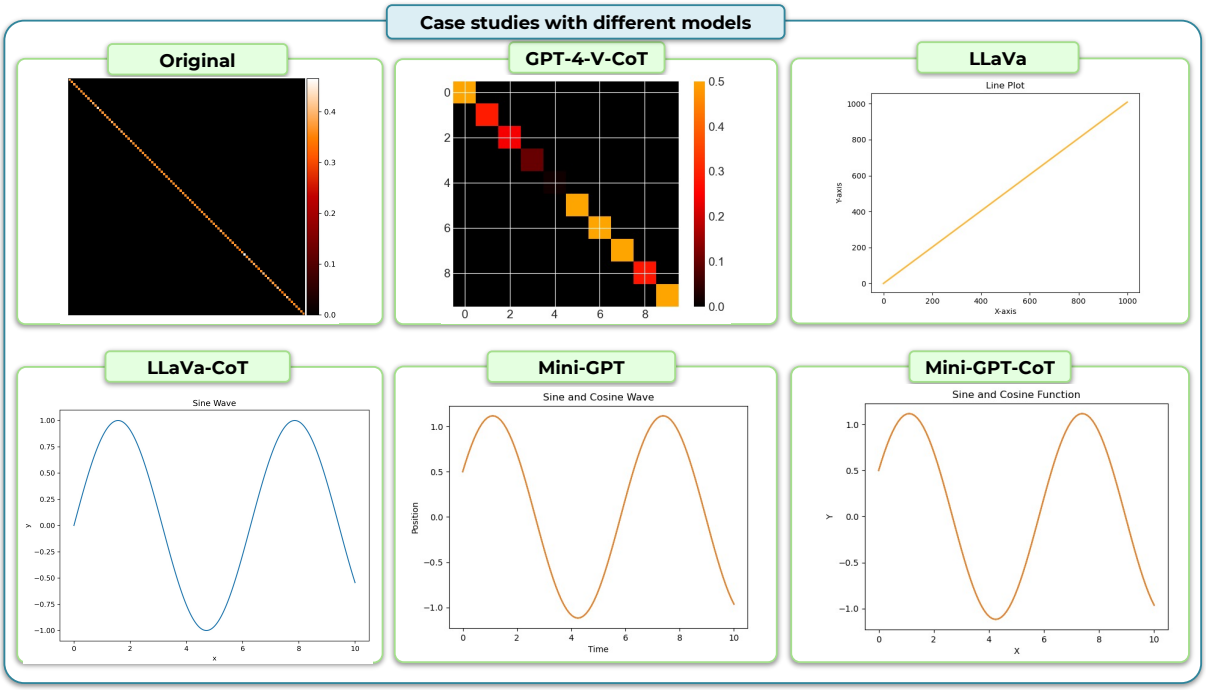



Figure 13: More case studies

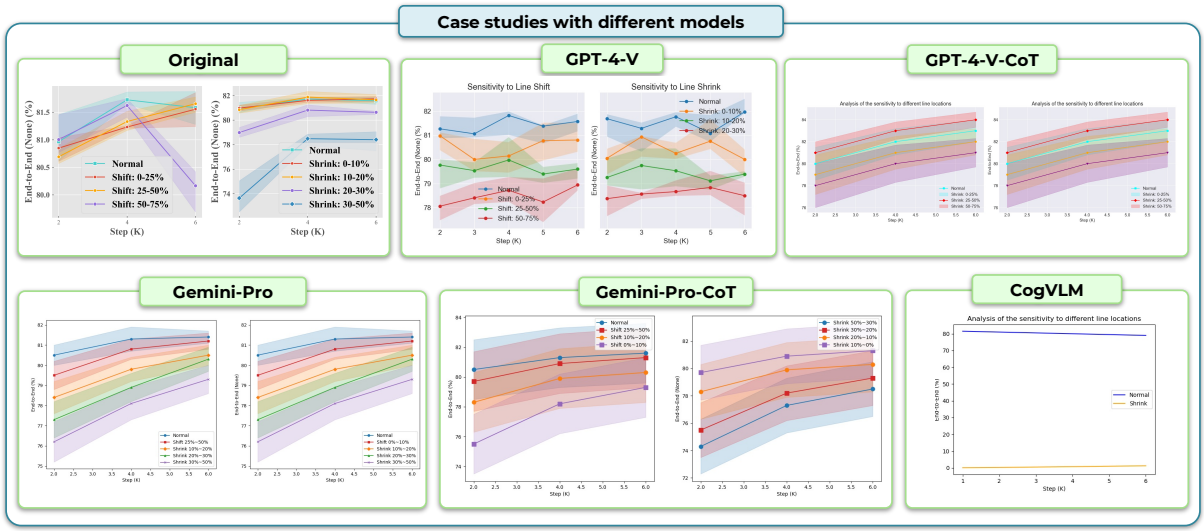



Figure 14: More case studies

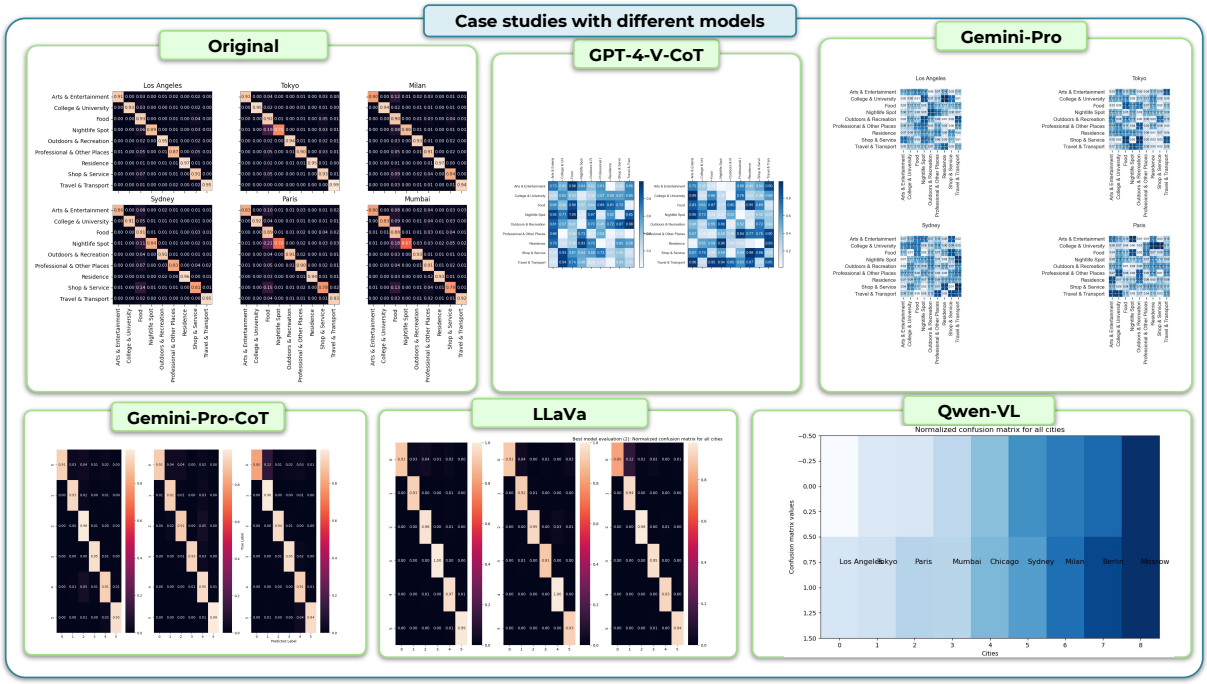
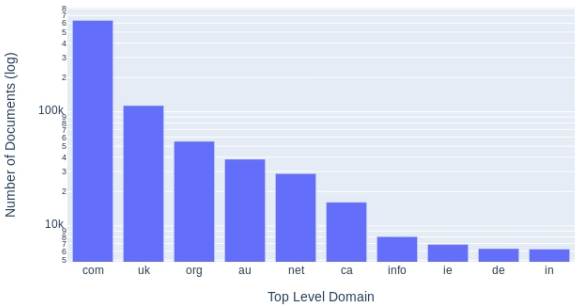
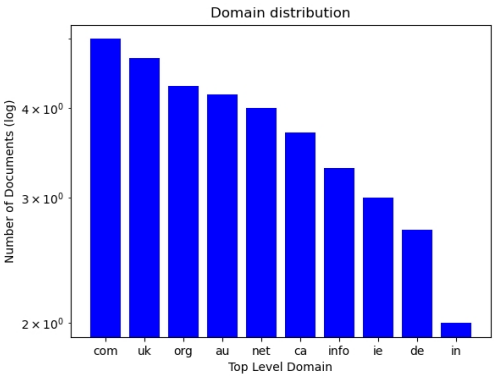



Figure 15: More case studies

**Original Figure**




**Generated Figure**




## Structural Components

Does the generated figure represent the correct type

- ◯ Yes
- ◯ No

Presence and placement of axes rating:

- ◯ 1 - Axes are missing or misplaced.
- ◯ 2 - Axes are present but not positioned accurately.
- ◯ 3 - Axes are mostly accurate but with minor positioning or scale issues.
- ◯ 4 - Axes are accurately placed with only negligible discrepancies.
- ◯ 5 - Axes placement is accurate and indistinguishable from the original.

Correctness of tick marks and grid lines rating:

- ◯ 1 - Tick marks and grid lines are missing or incorrectly placed.
- ◯ 2 - Some tick marks or grid lines are present but have significant inaccuracies.
- ◯ 3 - Most tick marks and grid lines are correctly placed but some errors are noticeable.
- ◯ 4 - Tick marks and grid lines are well-placed with very minor deviations.
- ◯ 5 - Tick marks and grid lines are placed exactly as in the original.

Rate the accuracy of text elements like titles, axis labels, legend, and annotations for style and position:

- ◯ 1 - Text elements like titles, axis labels, legend, and annotations are missing or completely different in style and position.
- ◯ 2 - Text elements like titles, axis labels, legend, and annotations are present but style and position poorly match the original.
- ◯ 3 - Text elements like titles, axis labels, legend, and annotations have a somewhat similar style and position but with notable differences.
- ◯ 4 - Text elements like titles, axis labels, legend, and annotations style and position are closely matched to the original with minor deviations.
- ◯ 5 - Text elements like titles, axis labels, legend, and annotations matche the original in both style and position perfectly.

Figure 16: The complete interface of our human evaluation (Page 1)

## Stylistic Components

Rate the use of color palettes and their matching with the original figure:

- ◯ 1 - Colors used are completely different from the original.
- ◯ 2 - Colors somewhat resemble those in the original, but the match is poor.
- ◯ 3 - Colors are generally similar, with a few inaccuracies.
- ◯ 4 - Color palette is very close to the original with negligible differences.
- ◯ 5 - Color match is perfect, with indistinguishable differences from the original.

Rate if the line types, bar types (solid, dashed, dotted, etc.), marker styles are consistent with the original:

- ◯ 1 - Line/bar/marker styles are inconsistent with no match to the original.
- ◯ 2 - Line/bar/marker styles show an attempt at consistency, but there are significant mismatches.
- ◯ 3 - Line/bar/marker styles are mostly consistent, with a few noticeable discrepancies.
- ◯ 4 - Line/bar/marker styles match well with the original, with minor inconsistencies.
- ◯ 5 - Line/bar/marker styles are consistent and match the original exactly.

## Numerical Components

Estimate the visual accuracy of numerical representations (e.g., bar heights, point locations) compared to the original:

- ◯ 1 - Numerical values are not at all accurately represented; major discrepancies are visible.
- ◯ 2 - Some elements are somewhat accurate, but there are significant visual differences.
- ◯ 3 - Most numerical values appear to be visually similar, with some minor inaccuracies.
- ◯ 4 - Numerical values are very closely represented, with very few and hard-to-notice differences.
- ◯ 5 - Numerical values are visually indistinguishable from the original figure.

## Practical Utility

Rate how helpful the figure is for easily recreating a similar style with minor modifications.

- ◯ 1 - It would be extremely difficult; the figure requires major revisions to be usable.
- ◯ 2 - It would be somewhat difficult; the figure needs several significant changes to be adaptable.
- ◯ 3 - It would be moderately easy; the figure needs some adjustments to be practical for reuse.
- ◯ 4 - It would be very easy; the figure requires only minor tweaks to adapt to new data.
- ◯ 5 - It would be extremely easy; the figure can be used as-is or with minimal modifications.

Figure 17: The complete interface of our human evaluation (Page 2)