Automating Academic Visualization: Harnessing Vision-Language Models for Reproducing Academic Data Visualization Figures

Zhehao Zhang, Max Lan & Qichao Wang
Department of Computer Science
Dartmouth College
Hanover, NH 03755, USA
{zhehao.zhang.gr, max.h.lan.gr, qichao.wang.gr}@dartmouth.edu

Abstract

This work explores the capability of state-of-the-art Vision Language Models (VLMs) in generating Python code to replicate data visualization figures from top AI conference papers. As data visualization becomes increasingly crucial in academic research, the challenge lies in efficiently replicating these visualizations for further analysis. To bridge this gap, we introduce AcademiaGraph, a dataset of diverse data visualization figures from leading AI conferences, and analyze the ability of SOTA VLMs to reproduce these figures through code generation. Our study focuses on models like GPT-4-Vision-Preview (ChatGPT-V) and LLaVa, assessing their performance across various criteria such as plot type accuracy, text elements accuracy, color usage, and overall utility for practical applications. The results indicate that while open-source VLMs face challenges in accurately reproducing these figures, ChatGPT-V shows a high proficiency level. This study highlights the potential of ChatGPT-V as a tool for enhancing data visualization practices in academic research, marking a significant step forward in the application of VLMs in academic settings.

1 Introduction

In the dynamic world of academic research, the role of data visualization has become increasingly vital for presenting complex information succinctly. Despite its importance, replicating these visualizations for further analysis or comparative studies poses significant challenges. Often, this requires extensive time and specialized skills in data management and graphic design. For instance, replicating a complex genomic data visualization or a multi-variable climate change model often demands substantial time and expertise in both data science and graphic design. It sometimes takes plenty of time for researchers to find relevant documents of Python libraries such as Matplotlib Hunter (2007) or Seaborn Waskom (2021). This barrier not only hinders knowledge dissemination but also limits the collaborative spirit essential in academia.

The emergence of large-scale Vision Language Models (VLMs)(Radford et al. (2021); Li et al. (2022); Liu et al. (2023)), introduces intriguing possibilities in this realm. Recently, GPT-4V(ision) OpenAI (2023a) is released and attracts immediate attention from the community for its outstanding multimodal perception and reasoning capability. Its superiority and generality are showcased in Yang et al. (2023). Although these models demonstrate a promising understanding of both images and text, their application in academic data visualizations remains largely unexplored.

To address this gap, this project aims to explore and leverage the ability of these Vision Language Models to bridge the gap between visual data interpretation and automated code generation. Specifically, we seek to answer the question: Can state-of-the-art VLMs generate appropriate code to reproduce an exact data visualization figure from an academic paper? We intend to assess how well these VLMs perform this task, identifying their strengths and

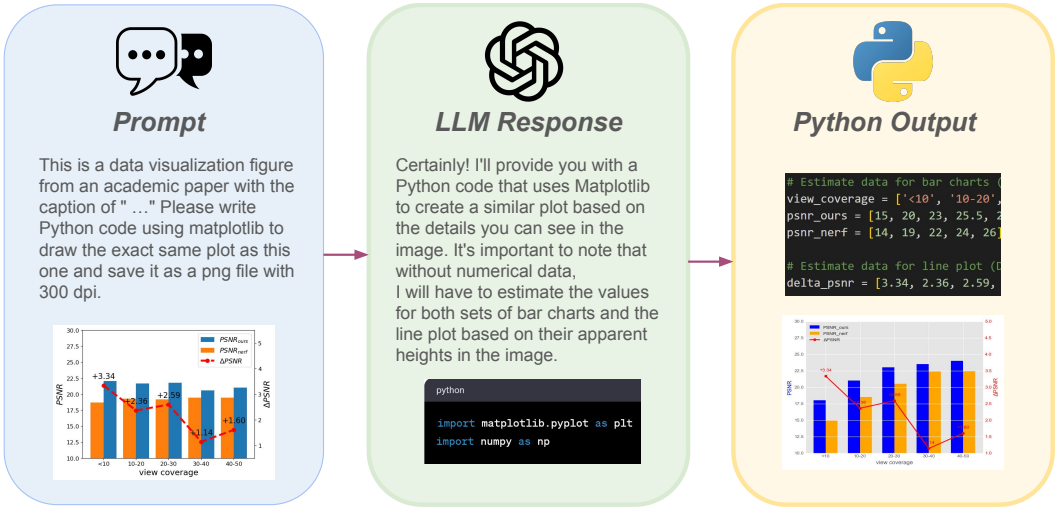

Figure 1: Our Task Formulation: We input instructional prompts from Vision-Language Models (VLMs) along with data visualization figures from top conferences, asking the models to write Python code that can reproduce the same figures. This study explores the effectiveness of current state-of-the-art (SOTA) VLMs in generating appropriate code for reproducing these figures.

weaknesses. We believe our project will shed light on how current VLMs can facilitate the research process and enhance the efficiency and accuracy of data representation in academic research.

Our contributions are as follows:

- We propose AcademiaGraph, a dataset consisting of a diverse range of data visualization figures in recent AI top-conference papers. Additionally, our data collection methodology is versatile and can be adapted to various domains and timeframes.
- We are the first to systematically analyze the ability of state-of-the-art (SOTA) VLMs to reproduce data visualization figures through code generation. We have developed a comprehensive evaluation protocol to assess the figures generated by their code from multiple perspectives. Moreover, we explore and compare the performance of these VLMs when enhanced with various prompting strategies.

2 Task Description

The task we focus on in this project is to leverage a VLM to interpret academic data visualizations and generate corresponding code for their reproduction. The task can be succinctly outlined as follows:

Given:

1. A data visualization figure F from an academic paper, accompanied by its caption C.
2. An instructional prompt P directing the model to generate code to replicate F.

The Vision-Language Model 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.

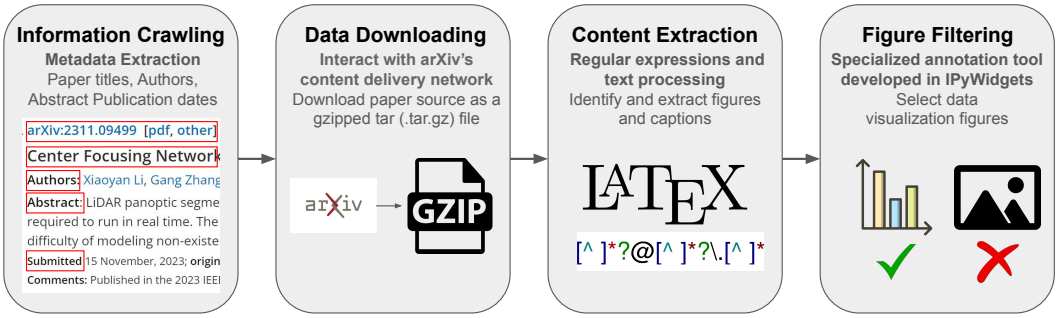

Figure 2: Data processing pipeline.

- 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, Seaborn). The task is formally expressed as:

$$S = M(F, C, P)$$

We visualize the task formulation in Figure 1.

3 AcademiaGraph Dataset

This section provides an overview of the data collection process employed for extracting images and captions from academic papers. The purpose of this effort is to gather a diverse dataset for several prestigious conferences in the field of artificial intelligence including data mining, machine learning, computer vision, natural language processing, and robotics.

3.1 Data collection

Our data collection process was meticulously designed to ensure both the integrity and relevance of the dataset for research applications. The initial phase, termed 'raw data scraping,' involved automated retrieval of academic papers from various computer science conferences. Following the acquisition of these raw papers, we proceeded to the 'figure filtering' stage. This crucial step was to filter out figures which are not data visualizations. Subsequently, the final phase of our data collection was 'data annotation.' This stage entailed a manual review process, pivotal for verifying the accuracy of the extracted figures and their corresponding captions. In the ensuing sections, we will continue to discuss the analysis and statistical insights derived from this dataset.

3.1.1 Raw data scraping

The raw data scraping process commenced with the development of a Python-based script, which programmatically interfaced with the arXiv API. The script was designed to query and retrieve metadata from the arXiv repository, focusing on papers published in the targeted conferences. The raw data scraping follows a sequential approach:

1. Querying arXiv with specific conference names and date ranges.
2. Downloading papers and extracting their source code or PDFs.
3. Parsing the source code (Latex) of each paper to identify figures and captions.
4. Storing the extracted data in a structured format for further analysis.

Information crawling and Parsing Using BeautifulSoup, we parse HTML content from arXiv to extract paper metadata and extract relevant information such as paper titles, authors, and publication dates. This module filters papers based on conference names.

Data Downloading We handle the downloading of papers using the extracted metadata, interact with arXiv's content delivery network, and retrieve papers in Latex format including tar and zip archives for source materials.

Content Extraction We identify and extract figures and captions from the downloaded papers. The core of the process lies in extracting figures and their respective captions. Regular expressions and text processing techniques are employed to accurately identify and separate figures and captions from the textual content of the papers.

The application of the script can yield a dataset of significant volume and variety.

3.1.2 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 method to input the figure into the open-sourced LLava model Liu et al. (2023) to determine if the figure is a data visualization figure. The prompt is as follows: "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." We also ensure that the figure represents data visualization during our manual copying and pasting of codes from the ChatGPT interface.

3.2 Dataset Analysis and Statistics

Our comprehensive dataset includes 332 data points, carefully curated from a range of prominent AI conferences. These conferences, known for their significant contributions to the field, encompass the 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), and Neural Information Processing Systems (NeurIPS).

Each data point in our dataset is structured with key elements to ensure thorough representation and analysis. These elements include:

- figure_path: The path to the figure within the dataset.
- caption: A descriptive caption of the figure.
- source: The source file related to the figure.
- arxiv_id: A unique identifier correlating to the figure's original submission or publication.
- type: The type of figure, categorizing the visual representation method used. caption: A descriptive caption of the figure.

The diversity of figure types in our dataset underscores the wide range of visualization techniques in AI research. The figure type distribution is as follows: Line Charts represent 51.91%, Scatter Plots are at 12.02%, Heat Maps make up 5.46%, Bar Charts are 22.95%, Histograms account for 2.73%, Box Plots are 3.28%, and Other figure types comprise 1.64%. This distribution highlights the rich variety of data visualization methods used in AI research, from conventional graphs to more specialized forms.

In summary, the Academic Graphs Dataset we propose serves as a vital resource for analyzing data visualization trends and methodologies in AI research. Its comprehensive nature and diversity make it an invaluable tool for studies focused on graphical analysis and the presentation of information in the AI domain.

4 Experiment

4.1 Models and baselines

4.1.1 GPT-4-Vision-Preview (ChatGPT-V)

GPT-4-Vision-Preview(OpenAI, 2023b), an extension of the GPT-4 model by OpenAI, is designed to integrate vision and language processing capabilities. This model extends the conventional text-based large language models by incorporating the ability to understand and generate responses based on visual inputs. In our context, GPT-4-Vision-Preview is utilized to directly generate Python code using matplotlib for data visualization.

Direct Prompting(Baseline) The baseline approach in our study is using a straightforward prompt for generating Python code with matplotlib, based on a figure's caption from an academic paper.

The prompt format is: *"This is a data visualization figure from an academic paper with the caption of '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."*

This method serves as a baseline, allowing us to measure the efficacy of more complex methods against a simple and direct approach.

Chain-of-Thought Prompting (CoT) Chain-of-Thought prompting introduces a method of enhancing the reasoning abilities of large language models through a series of intermediate reasoning steps(Wei et al., 2022). The approach augments standard few-shot prompting with a chain of thought for each exemplar, significantly improving performance on tasks requiring complex reasoning, arithmetic, commonsense, and symbolic reasoning tasks(Wei et al., 2022). We implement COT through extending the direct prompting through adding *"Let's think step by step."* at the end of the message.

4.1.2 LLaVa

LLaVa (Large Language and Vision Assistant) Liu et al. (2023) is an end-to-end trained large multimodal model that connects a vision encoder and a large language model (LLM) for generalized visual and language understanding. The architecture of LLaVa effectively leverages the capabilities of a pre-trained LLM and a visual model, such as LLaMA for language understanding and CLIP's ViT-L/14 for visual encoding. The training involves generating multi-turn conversation data for each image, and the sequences are organized to treat all answers as the assistant's response. We investigate the data visualization generation task on the LLaVa model the same as the baseline method.

4.2 Evaluation protocol

As there is no automatic metric to evaluate our task, we use human evaluation in this work. Our evaluation protocol for assessing the figures generated by VLMs is detailed across several key dimensions:

- **Plot Type Accuracy**: Assessing if the generated figure correctly represents the plot type described in the source material. This includes verifying the fidelity in representing various chart forms like bar graphs, line charts, scatter plots, etc.
- **Axes Evaluation**: Focusing on the presence, correct placement, and alignment of axes. The criteria range from the absence or misplacement of these elements to their perfect replication from the original figure.
- **Tick Marks and Grid Lines Evaluation**: Examining the accuracy of tick marks and grid lines. The assessment spans from complete absence or incorrect placement to a precise match with the original figure.
- **Text Elements Accuracy**: Evaluating the replication of text elements such as titles, axis labels, legends, and annotations, both in terms of style (font type, size) and positional accuracy.

- **Color Usage Evaluation**: Checking how closely the colors in the generated figure match those in the original, including shade, intensity, and overall palette consistency.
- **Line Styles Consistency**: Assessing the consistency of line, bar, and marker styles compared to the original figure, focusing on the accuracy of these stylistic elements.
- **Numerical Value Similarity**: Evaluating the visual accuracy of numerical representations, like the height of bars in a bar chart or the position of points in a scatter plot, ranging from major discrepancies to an exact match.
- **Figure Adaptability for Practical Use**: Assessing how easily the generated figure can be adapted for practical applications, ranging from requiring significant modifications to being ready for use with minimal or no changes.

We have developed an interface using Ipywidgets that allows users to rate the aforementioned perspectives on a scale of 1 to 5, with the exception of plot type accuracy, which is assessed as a 'yes' or 'no' question.

4.3 Experiment setups

For the ChatGPT-V baselines, we manually queried the system through the ChatGPT web interface OpenAI (2023). For LLava Liu et al. (2023), we utilized the 8-bit quantized version of the llava-v1.5-13b checkpoint. All experiments with open-sourced Vision Language Models (VLMs) were conducted using an Nvidia RTX 3080 Laptop with 16 GB of GPU memory. We set the hyperparameter of temperature to 0.2 and max_new_tokens to 512.

5 Results

Method	Type (ACC)	Axis	Marks&Grid	Text	Color	Style	Number	Utility
Direct Instructional Prompting								
LLaVa	0.51	2.00	1.82	1.35	1.49	1.88	1.18	1.67
ChatGPT-V	0.87	2.89	2.88	2.95	2.74	2.85	2.50	2.73
Zero-shot Chain-of-Thought								
ChatGPT-V	0.95	3.48	3.35	3.30	3.03	3.35	2.57	3.51

Table 1: Human evaluation results on different baselines' performances from different perspectives.

We present the experiment results in Table 1. It is evident that compared with LLaVa, ChatGPT-V greatly outperforms it in all respects, indicating that ChatGPT-V is significantly superior to the SOTA open-source VLM for our task. Notably, the Utility value for LLaVa is extremely low (1.67), suggesting that LLaVa is largely ineffective in reproducing data visualization figures and can hardly be utilized in this task. In contrast, ChatGPT-V scores well in multiple aspects, particularly in reproducing marks & grids, and text elements. However, the model is less capable of replicating colors. Among all the different aspects, all models perform the worst in replicating numerical features, which is reasonable since the original data is not provided in the prompt. This, however, does not significantly impact its utility, which is the most crucial metric for practical usage. Comparing the performance of direct prompting with zero-shot Chain of Thought (CoT) prompting, we find that CoT can noticeably improve the ratings in all aspects, indicating that this simple yet effective prompting technique remains valuable for VLMs in our task. Additionally, we compared the results produced by Zero-shot CoT and direct prompting. According to human evaluation, in 34.83% of cases, direct prompting outperformed Zero-shot CoT; in 37.08% of cases, Zero-shot CoT was better than direct prompting; and in 28.09% of cases, they performed equally well.

6 Case Study

Apart from quantitative experiments, we also conduct comprehensive qualitative case studies to get a deeper insight into ChatGPT-V in our task.

Criteria	Case 1	Case 2	Case 3	Case 4
Plot Type Accuracy	5	3	1	X
Axes Evaluation	5	4	1	X
Tick Marks and Grid Lines Evaluation	5	5	2	X
Text Elements Accuracy	3	3	1	X
Color Usage Evaluation	5	5	3	X
Line Styles Consistency	5	5	4	X
Numerical Value Similarity	4	2	1	X
Figure Adaptability for Practical Use	5	3	1	X

Table 2: Example Evaluation for Case Study

6.1 Example Case 1 - A Good Replication

In Figures 3 and 4, the original image (Figure 3) consists of a combined pie chart and bar chart, demonstrating some complexity, yet it remains a basic visualization. The model exhibits a strong ability to capture the textual content and data patterns in the original plot. The newly generated plot closely resembles the original, with only minor differences in the orientation of the pie chart. It is important to note that since the original plot was not created using matplotlib, while our model generates plots using this library, some stylistic differences are inevitable. Overall, the axes and text are accurately represented, and the display format is consistent. The numerical values (height of the bars and proportion of the pie) are almost identical. The model attempts to replicate the same colors, achieving this with notable accuracy, except for a minor discrepancy at the rightmost part of the bar chart, likely due to the inherent randomness of the language model.

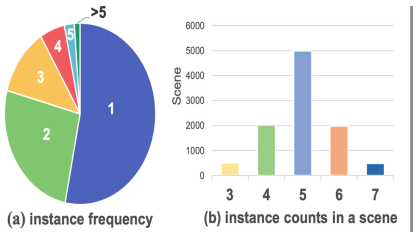

Figure 3: Example-1-Original Image

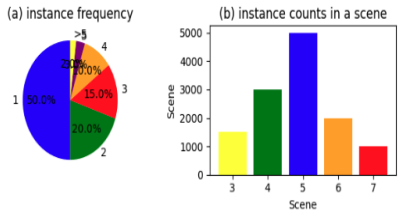

Figure 4: Image Generated by Code

In Figures 5 and 6, the original image (Figure 5) consists of four line charts with relatively simple patterns. However, the generated plot does not accurately place the caption. It captures the color and texture of the lines, as the model is typically proficient in reproducing correct colors and line or bar structures. Despite this, as demonstrated in this graph, the model struggles with more complex patterns and shapes, often resulting in poor numerical accuracy. For instance, in this example, although the lines appear similar at a glance, their shapes are almost inversely replicated compared to the original.

6.2 Example Case 3 - Example of Failure

In Figures 7 and 8, the original image (see Figure 7) is a complex three-dimensional scatter plot with a spherical reference shape. Given the relative rarity of this plot type (compared to more common visualization methods) and the added complexity of the reference shape, our model struggles to comprehend the dimensions and spatial arrangement. The axes are completely missing, no text is included, and the plot lacks numerical accuracy.

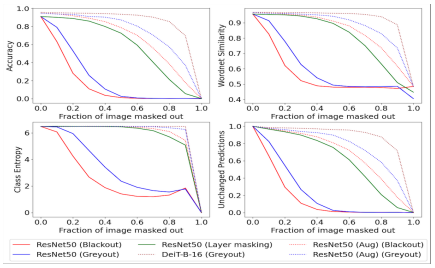

Figure 5: Example-2-Original Image

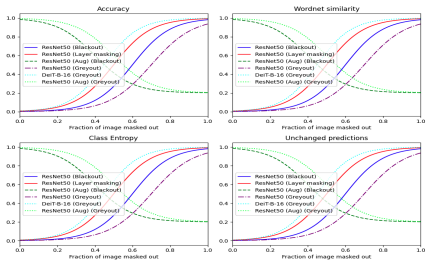

Figure 6: Image Generated by Code

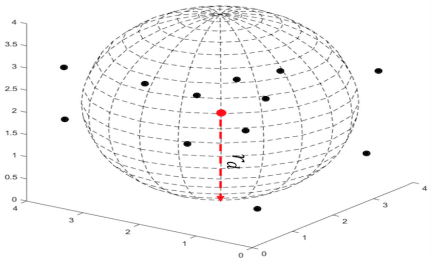

Figure 7: Example-3-Original Image

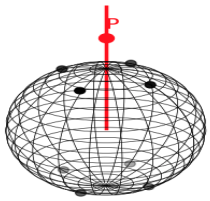

Figure 8: Image Generated by Code

7 Conclusion

In this study, we conducted a comprehensive analysis of the capability of current state-of-the-art (SOTA) Vision Language Models (VLMs) to replicate data visualization figures found in top AI conference papers. To facilitate this analysis, we developed AcademiaGraph, a diverse dataset encompassing a wide range of data visualization figures from recent leading AI conferences. Our systematic evaluation of SOTA VLMs, focusing on their code generation capabilities, reveals a notable disparity: while the best open-source VLMs struggle to effectively reproduce these figures, ChatGPT-V demonstrates a generally high level of proficiency in this task. This finding not only underscores the advanced capabilities of ChatGPT-V but also highlights its potential as a valuable tool for researchers seeking to streamline the process of data visualization. We are optimistic that our work will significantly contribute to simplifying and enhancing data visualization practices in academic research.

References

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

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

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

OpenAI. Chatgpt web interface. `https://chat.openai.com`, 2023. Accessed: 2023-11-13.

OpenAI. Gpt-4 technical report, 2023a.

OpenAI. Gpt-4: Openai's multimodal language model. `https://openai.com/blog/gpt-4`, 2023b. Accessed: 2023-11-13.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning

transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6 (60):3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

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

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