# Reproduce Work in Advancing MLLMs in ChartQA with Visulization-Referenced Instruction Tuning

**Abstract**

Traditional Chart QA often follows a two-stage process: first, converting a chart into a data table using a Chart-To-Table model, and then using a Large Language Model (LLM) to generate answers based on the table and the user's question. However, this method has limitations, such as the loss of visual details like color and layout during the conversion. Some studies address this by improving training data quality through data augmentation, but these approaches often rely on frozen visual encoders, which struggle to adapt to detailed chart features. To overcome these challenges, the authors introduce a new data engine that creates 200,000 high-quality chart QA examples through data filtering and synthesis, followed by instruction-tuning on the LLaVA-HR model with unfrozen parameters.

In my reproduce of this work, I first reproduced the original paper's results, achieving similar performance on the same datasets. I then assessed the model's accuracy across different question types and identified weaknesses in numerical computation tasks. To improve this, I integrated a numerical computation library. For questions requiring numerical computation, the model generates a structured outline with computation types and data, processes this through the library for precise results, and then provides the final answer to the user.

**Keywords:** ChartQA, MLLMs, Visulization, Data Engine.

## 1 Introduction

The ability to interpret and reason about visual data (especially diagrams) is a key skill in various fields such as data analysis, scientific research and business intelligence. Multimodal Large Language Models (MLLMs) have shown great potential for understanding and generating textual content, but they still perform poorly in tasks involving graph comprehension. This limitation is particularly evident in tasks that require numerical computation and complex reasoning based on visual data. Since graphs are a common medium for presenting data, improving the ability of MLLMs to understand and reason about graphs is critical to improving their use in real-world applications.

Recent developments in MLLMs have focused on two approaches to improving diagram comprehension: model-centered approaches and data-centered approaches. Model-centered approaches aim to enhance the architecture of the model, such as improving visual coders and projectors to better capture the unique features of diagrams. On the other hand, data-centric approaches focus on enhancing the quantity and quality of training data to improve the performance of the model. For example, models such as ChartLlama [1] utilize large-scale

instruction-based datasets generated by high-level language models such as GPT-4 [2] to fine-tune their visual encoders. However, these models typically rely on encoders pre-trained on natural images that have difficulty handling the unique visual characteristics of charts.

Chart Question Answering (ChartQA) datasets play a pivotal role in training and evaluating models for interpreting chart data. These datasets typically include various types of charts, such as bar, line, and pie charts, paired with questions that require reasoning about the presented visual and numerical data.Well-known datasets such as PlotQA [3] and ChartQA [4] have been instrumental in advancing the field by providing a variety of question types and chart formats. Despite the contributions of existing datasets, they face several challenges such as uneven data distribution, limited representation of real-world scenarios, and inefficient integration of large-scale data.

To address these challenges, Zeng et al. proposed a novel data engine [5] that combines data filtering and generative techniques to create a high-quality, diverse dataset for training MLLM. The data filtering module ensures a balanced presentation of chart types and attributes, while the data generation module introduces a robust, context-oriented approach to enhance the diversity and relevance of the dataset. In addition, they explored unfrozen visual coders combined with mixed resolution adaptation strategies to improve the model's ability to recognize fine-grained attributes of charts.

In my replication work, I first developed a web-based end-to-end chart quiz bot, which I then evaluated on the ChartQA dataset, obtaining results largely consistent with the original article. I further wanted to explore the differences in the model's accuracy in answering different question types, and got that the model was less accurate on questions involving numerical calculations. I therefore introduced an additional numerical computation library meant to decouple the process of reasoning and computation of the model.

## 2 Related works

### 2.1 ChartQA Methods

**Chart Question Answering** involves tasks where models are required to respond accurately to questions derived from chart content. This task challenges models to identify trends within the data and discern relationships between various data points. Research in this area has focused on two primary categories of questions: factoid and open-ended. Factoid questions typically result in concise answers such as nouns (e.g., axis values), verbs (e.g., indicate increases or decreases), or adverbs (e.g., describing the magnitude of a trend). In contrast, open-ended questions necessitate more elaborate responses, often expressed in complete sentences [3–5].

Researchers have continually pushed the capabilities of vision-language (VL) models, particularly in the area of Chart Question Answering (CQA). This research can primarily be divided into two methods.

The first involves a two-step process: first, vision models turn charts into data tables, and then language models interpret them [4, 6, 7]. However, this approach often struggles to keep visual information, like color and layout, intact, which limits where it can be effectively used [7].

The second method relies on integrated VL models that handle charts and text data together in one step [8–10]. Notable models such as Matcha [8] improve an existing general VL model, Pix2Struct [11], adding the ability to perform mathematical operations and extract data from charts. This results in better performance in CQA and the captioning. Similarly, UniChart [9] broadens Matcha's scope by using a larger dataset to prepare

the model for a wider range of chart-based tasks. Still, these models sometimes fall short in their handling of language, particularly with tasks that require complex calculations [4].

The rise of Multimodal Large Language Models (MLLMs) has shifted the landscape, leading to significant progress in visual question answering. The InternVL2.5 model [12], an open-source MLLM, surpasses other chart-specific models in the ChartQA evaluations [4]. Notably, InternVL2.5 excels at answering questions posed by people, rather than those generated by computers.

### 2.2 Enhancing MLLMs' Performance in Chart Understanding

The work on enhancing multimodal large models (MLLMs) for chart question answering can be broadly categorized into two directions: model-centered approaches and data-centered approaches. Model-centered methods focus on improving the architecture of the model to boost its performance, such as enhancing the efficacy of visual encoders and projectors. On the other hand, data-centered efforts aim to improve the performance of MLLMs on chart understanding tasks by augmenting the quantity and quality of the training data. For instance, ChartLlama [1] generates 160K instruction-based data using GPT-4 and fine-tunes these instructions on a frozen visual encoder. However, these models are predominantly built on CLIP encoders pretrained on natural images, which face challenges in handling the unique visual features inherent to charts.

### 2.3 ChartQA Datasets

Chart Question Answering (ChartQA) datasets are essential resources for training and evaluating models that are designed to answer questions based on chart images. These datasets typically contain various types of charts (e.g., bar charts, line graphs, pie charts), paired with questions and answers that require reasoning about the visual data presented in the charts. They have played a crucial role in the development of multimodal models that integrate both visual and textual understanding. Below are two prominent ChartQA datasets:

The PlotQA [3] dataset is one of the earliest and most comprehensive resources created for chart-based question answering. It includes a large collection of charts, such as bar charts and line graphs, with questions that vary in difficulty and complexity. The dataset contains over 100,000 questions, covering a broad range of question types, from simple factual queries (e.g., "What is the value of X?") to more complex reasoning tasks (e.g., "How does the value of X change over time?"). The dataset pairs each chart with corresponding questions and answers and aims to provide a balanced selection of common chart types alongside challenging visual reasoning tasks. Notable features include annotations of both visual and textual elements, a wide variety of question-answer pairs, and suitability for training and evaluating models on both simple and complex ChartQA tasks.

The ChartQA dataset [4], developed by Bhattacharya et al., is designed to assess models' abilities to understand and reason about chart data. It includes a diverse set of question types that require models to perform reasoning tasks on bar charts and line graphs, such as comparisons of values, data point identification, and trend analysis. ChartQA contains more than 6,000 questions associated with 1,000 chart images, covering a wide range of data-driven queries. The dataset is intended to test models' understanding of both the visual and numerical aspects of chart data. Key features of the dataset include a focus on charts with distinct visual elements, an emphasis on reasoning over charts rather than simple data retrieval, and the inclusion of questions that require multi-step reasoning or comparisons across multiple data points.

# 3 Method

## 3.1 Overview

## 3.2 Data Engine

The current Chart Question Answering (CQA) datasets face several challenges, including uneven data distribution, limitations in large-scale data integration, and differing data requirements across models. Empirical research indicates that existing datasets exhibit distributional flaws, failing to adequately represent the diverse chart features and question-answer tasks encountered in real-world scenarios. Furthermore, integrating all available CQA data without precise annotations results in large datasets that are inefficient to learn from and incur high training costs. Additionally, there are significant differences in the data requirements of different models. For instance, LLaVA [13] requires only 1,223K instruction data, while UniChart [9] and ChartAssistant [10] use 6,900K and 39.4M chart-related data points, respectively. These variations suggest that merely increasing the volume of data is neither feasible nor necessarily effective for improving model performance.

The distributional and scale issues within existing datasets indicate that relying solely on them will not effectively address the training challenges of CQA tasks. Generating new, high-quality data is essential for improving the model's ability to adapt to real-world scenarios while also enabling better control over the data scale, ensuring both efficiency and relevance in training.

### 3.2.1 Data Filtering Module

The goal of the data filtering module is to select representative data from existing CQA datasets, balancing the distribution of chart types and their respective attributes. This ensures that the selected data captures the diversity of real-world scenarios while maintaining an appropriate scale to reduce training costs. The filtering process follows several principles: first, chart types are classified according to visual literacy research, which categorizes charts such as bar charts, line graphs, and pie charts; second, fine-grained chart attributes, such as color, trends, and layout, are defined based on recognized attributes from visual retrieval tasks; and third, key attributes that impact Multimodal Large Language Models (MLLMs) for chart understanding, including the presence of numerical annotations and data groupings, are carefully considered.

The image classifier uses a frozen ConvNeXt backbone network to classify chart types and other attributes. The classifier labels fine-grained attributes, which are then used for stratified sampling. The training data for this classifier is sourced from the chart data in [3, 14], supplemented with manual annotations for missing attributes. To address class imbalance in the dataset, a focal loss function is employed, which assigns higher weight to underrepresented categories, ensuring that the training process is not dominated by more common categories. The focal loss function is defined as:

$$\mathcal{L}_{FL} = -\alpha_t(1-p_t)^{\gamma} log(p_t) \tag{1}$$

The features extracted from the charts are concatenated into a unified feature space, and k-means clustering is applied to group the charts into k clusters based on these features. Within each cluster, stratified sampling is performed according to the predicted chart attributes to ensure a balanced distribution. To avoid redundancy, an undirected graph is constructed, connecting image pairs with cosine similarities above a threshold($\epsilon$). The

images with the lowest cosine similarity to the cluster centroids are retained to ensure diversity in the dataset. The threshold $\epsilon$ is adjusted to select approximately 69K charts, which helps control the training costs.

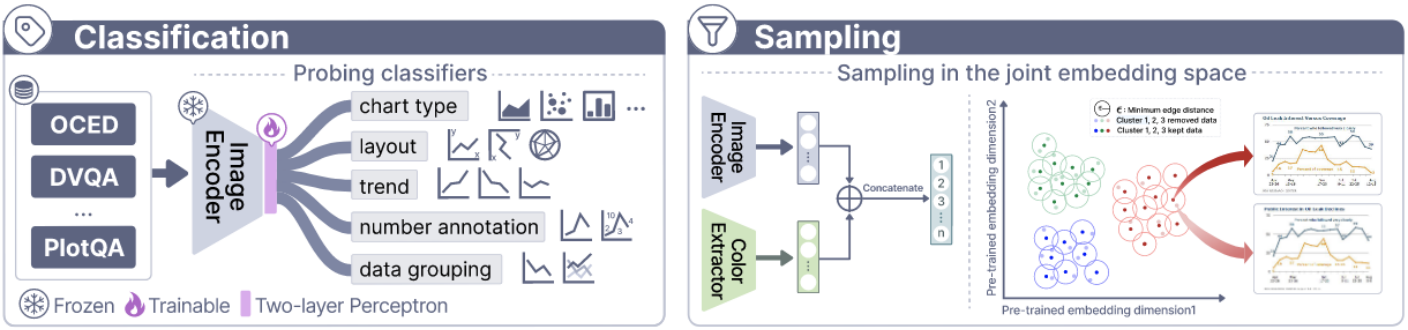


Figure 1. Illustration of the data filtering process, encompassing classification and sampling. Classification aims to investigate the distribution of existing datasets across key categorical attributes, including chart types, layout, trend, number annotations, and data grouping. Subsequently, we conduct sampling based on the fine category.

### 3.2.2 Data Generation Module

Existing methods often overlook the instability of language model outputs and fail to consider the importance of understanding the spatial structure of charts in guiding generation. To address this, the data generation module introduces more robust and contextually guided chart generation approaches.

The goal of RAG is to provide contextually relevant seed chart examples, which enhance the quality of generated charts. Seed chart collection involves gathering table-code pairs from authoritative chart libraries such as Vega-Lite, Matplotlib, Seaborn, and ECharts. These examples are further expanded by using RAG, which retrieves similar tables based on cosine similarity, constructing prompts for chart generation. Table features, including 30 cross-column features and 81 single-column features, describe the relationships between columns and column attributes. These features are used to match each collected table to the most similar table in the existing table-code pairs, enhancing the quality and accuracy of chart generation.

This approach aims to enrich the visual expressions of charts by adjusting visual elements such as numerical annotations, groupings, bar widths, and axis truncations or inversions. The goal is to generate diverse samples that expand the dataset's visual encoding diversity and better represent real-world data distributions.

### 3.2.3 Visualization-referenced Dataset

The new dataset covers 11 chart types and 8 task categories, containing 10,385 table-chart pairs and 51,245 chart-QA pairs. It incorporates data from various sources, resulting in a large-scale dataset with 199K data points, including 80K table-chart pairs and 119K chart-QA pairs. Compared to earlier datasets that rely on single data sources or templates, this dataset integrates chart data from multiple real-world and synthetic sources, offering greater diversity in visual encoding. It focuses on chart types and question categories that are often underrepresented in current datasets, such as range determination and distribution characterization. Moreover, it provides a more balanced representation of chart types and QA pair distributions.

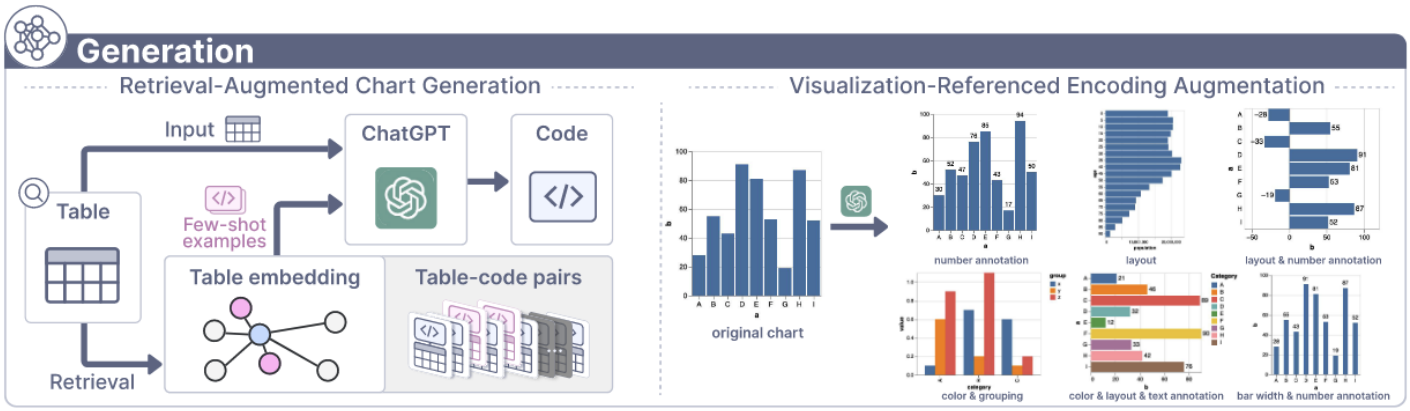


Figure 2. Data generation pipeline. First, we conduct retrieval-augmented chart generation with a set of table-code pairs we collected. This results in a collection of images distributed evenly in the real-world chart space. Then, we conduct visualization-referenced encoding augmentation for each seed chart to further enrich the dataset's size and diversity.

### 3.3 Model

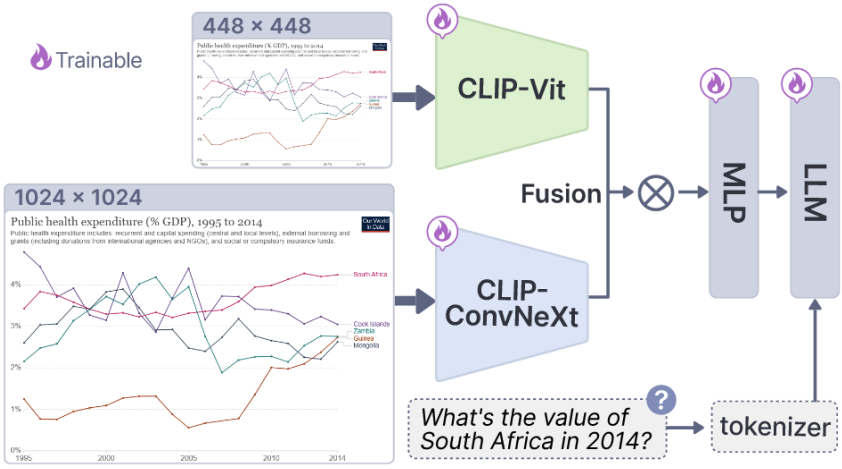


Figure 3. Architecture of the LLaVA-Chart. High-resolution and normal-resolution features of the input image are fused to facilitate the efficient recognition of fine-grained features. During the training phase, vision encoders are unfreezed to enable the adaptation to chart characteristics.

To improve the chart comprehension capabilities of Multimodal Large Language Models (MLLMs) in real-world scenarios, the authors propose two enhancements in both model architecture and training procedures. Specifically, they implement a mixture-of-resolution adaptation strategy [15] to improve fine-grained recognition. Additionally, to better capture the visual features of charts, they unfreeze the vision encoder during training and utilize the visualization-referenced dataset outlined in Section 3.2.3 for model training.

#### 3.3.1 Model Design

The authors utilize LLaVA-1.5 [13] as the core architecture for the model, which incorporates the CLIP-Vit-334px as the vision encoder, a two-layer Multi-Layer Perceptron (MLP) as the projector, and the Vicuna-13B [16] as the Large Language Model (LLM).

The two-layer MLP effectively bridges the feature space between the vision encoders and LLMs, but the resulting visual tokens are directly influenced by the image resolution. For example, CLIP-Vit-L-14 generates 5,329 tokens for a 1,022×1,022 resolution image, as each token corresponds to a 14×14 image patch [15], leading to significant computational overhead for MLLMs. A common strategy to handle high-resolution inputs, such as QFormer [17], requires extensive pre-training for achieving vision-language alignment, which is not feasible for visual tasks due to the scarcity of high-quality data. To address this, we adopt a resolution-adaptation strategy [15] that enhances the resolution while allowing training on standard scale data. This strategy integrates high-resolution features into low-resolution features using adapters, thereby reducing the number of visual feature tokens. Specifically, following the approach used in LLaVA-HR [52], we combine the CLIP-ViT-L [18] and CLIP-ConvNeXt [19] vision encoders using an adaptation strategy, which helps manage the length of the visual token sequence. The resolution settings for the ViT and CNN encoders are 448×448 and 1,024×1,024, respectively.

#### 3.3.2 Training Settings

Previous works [1, 13, 20] have opted to freeze the vision encoder during the entire training process, assuming that the pre-trained CLIP model is sufficiently effective at extracting features from natural images. Their training focus was on aligning the features extracted by CLIP to the LLM embedding space by fine-tuning the projector and LLM. However, previous research has shown that CLIP performs poorly on visualization images due to the limited number of charts and coarse annotations in its pre-training corpus, which results in suboptimal chart recognition capabilities. By unfreezing the CLIP parameters, the authors allow the model to adapt better to the specific characteristics of charts, thereby improving the overall chart comprehension performance of the MLLM.

The authors bypass the pre-training phase and directly leverage the initial projector weights from LLaVA-HR [15] for instruction tuning. Our objective is to enhance the general chart comprehension abilities of MLLMs. Therefore, our training dataset consists of two components: the 665K original instruction tuning data from LLaVA-1.5 [13] and the 199K chart-related data detailed in Section 3.2.3.

For optimization, we use the AdamW optimizer. The learning rate (LR) is set to 2e-5, and the global batch size is set to 128. The number of training epochs is set to 1, and the LR scheduler follows a cosine decay with a warmup ratio of 0.03. The training process runs on 16×NVIDIA A800 GPUs for approximately 19 hours.

## 4 Implementation details

### 4.1 Comparing with the released source codes

The differences compared to the published source code are: the former only discloses the code implementation of the model part and the training code, I supplemented the code for dataset loading as well as distributed training and evaluation, and support the use of different models and checkpoints for multi-gpu training and evaluation on multiple datasets. In addition, I introduce a numerical computation library for post-processing with the aim of decoupling model inference from computation to improve the accuracy of the model with respect to problems requiring numerical computation.

### 4.2 Reproduce of experimental results

In the evaluation session, we use the ChartQA dataset with QWen-VL-Chat and Chart-Llama as the baseline models, and the evaluation metrics use relaxed accuracy [3], i.e., 5% error is allowed for numerical answers, and exact match is required for textual answers.

Our reproduction results are comparable to the original results, as shown in the Table 1. It is worth noting that on the ChartQA-H dataset, our reproduction results are 1.6 points better compared to the original article for the following reasons: the original article used 16 A800s to complete the fine-tuning of the model commands in 1 day, and considering the cost of training, we used the author's publicly available checkpoints here, and I subsequently contacted the author, who said that the latest open checkpoint is the version with the best training effect.

Table 1. Reproduce results on the ChartQA dataset using different models vs. original results

| Model | ChartQA-M | | ChartQA-H | | Average | |
|---|---|---|---|---|---|---|
| | Reproduce | Paper | Reproduce | Paper | Reproduce | Paper |
| Qwen-VL-Chat | 85.36% | 85.36% | 47.32% | 47.32% | 66.34% | 66.34% |
| Chart-Llama | 90.40% | 90.36% | 48.40% | 48.96% | 69.40% | 69.66% |
| LLaVA-Chart | **93.44%** | **93.6%** | **65.28%** | 63.6% | **79.36%** | 78.60% |

### 4.3 Improvements

Since the question types in the ChartQA-H dataset can be categorized as data retrieval, visual, compositional, and visual-compositional, we would like to explore the model's answering accuracy on different question types, and the baseline models used here are Qwen-VL-Chat and GPT4-Vision-Preview, and the evaluation metric is still relaxed accuracy.

As shown in the Table 2, we can see that the model's accuracy in answering questions involving numerical computation is always unsatisfactory. Previous work has shown that large models are not good at the mathematical computation aspect of the problem, so the point of improvement we consider for the original work is that for questions that require numerical computation, we can first generate a structured skeleton, e.g., computation type: summation, data:... , then introduce an additional numerical computation library, input the structured data into the numerical computation library to get an accurate answer, and finally input that answer into the model for an accurate answer.

The improved modeling framework is shown in Figure 4. It can be compared with the original framework of the paper, as shown in Figure 3.

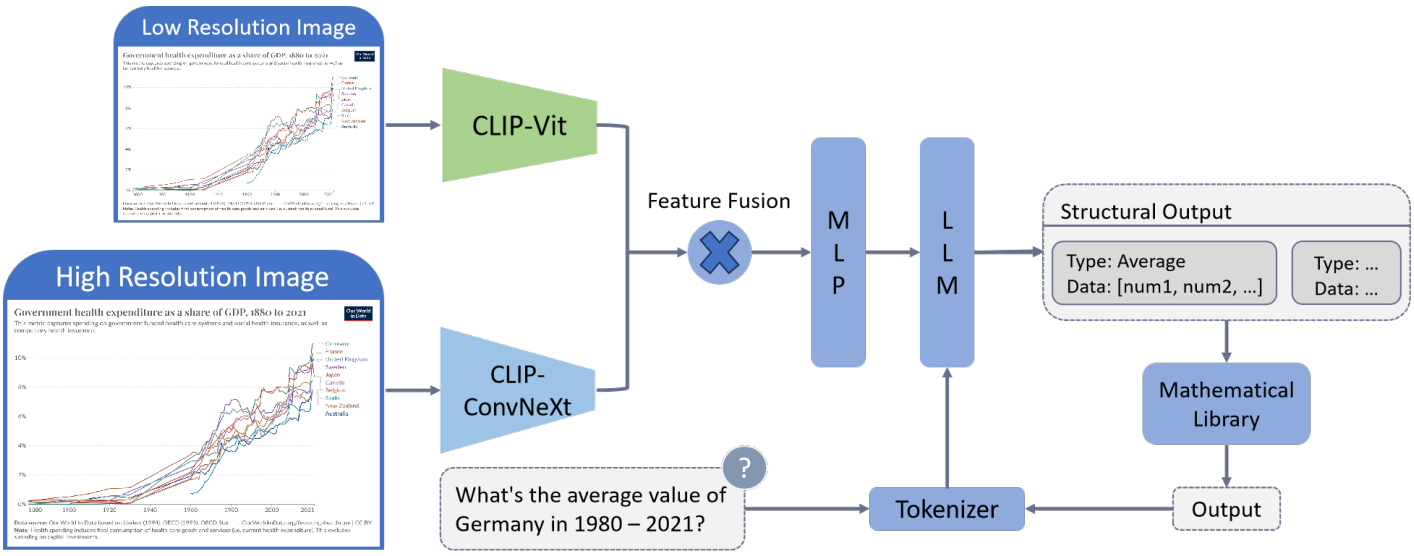


Figure 4. The model architecture of LLaVA-Chart after the introduction of additional numerical computation libraries. The structured skeleton is first generated by the model, then the numerical computation library is used to get accurate results, and finally the generated results are fed into the model to do a multi-round inference.

Table 2. Accuracy of model responses on different question types

| Model | ChartQA-M | ChartQA-H | | | | |
|---|---|---|---|---|---|---|
| | | Data Retrieval | Visual | Compositional | Visual-Compositional | - |
| Qwen-VL-Chat | 85.36% | 66.67% | 62.29% | 23.79% | 18.92% | |
| GPT4-Vision-Preview | 87.25% | 68.43% | 68.84% | 25.96% | 20.85% | |
| LLaVA-Chart | **94.44%** | **79.92%** | **74.49%** | **52.38%** | **46.84%** | |

Table 3. Accuracy of model responses on different question types

| Model | ChartQA-M | ChartQA-H | | | | |
|---|---|---|---|---|---|---|
| | | Data Retrieval | Visual | Compositional | Visual-Compositional | - |
| LLaVA-Chart | 94.44% | 79.92% | 74.49% | 52.38% | 46.84% | |
| Improved-LLaVA-Chart | 94.44% | 79.92% | 74.49% | **62.24%** | **58.36%** | |

## 5 Conclusion and future work

Traditional Chart QA typically takes a two-stage approach: first, the chart is converted to a data table through a Chart-To-Table model, and then the user's question and the data table are combined to provide a reasoned answer using a Large Language Model (LLM). However, this approach has some limitations. For example, during Chart-To-Table conversion, visual information (e.g., color and layout) of the chart is lost.

On the other hand, some research focuses on data augmentation, which improves the model's understanding of charts by enhancing the quality of the training data. The shortcoming of such approaches is that most of them rely on frozen visual coders for Instruction-Tuning, which cannot effectively adapt to the fine-grained features in charts. To address these issues, the authors propose a novel data engine that generates 200,000 high-quality chart QA data through data filtering and data synthesis, followed by instruction-tuning on the unfrozen LLaVA-HR model.

In my reproduce work, I first reproduce the experimental results of the original paper and achieve consistent performance on the datasets it uses. Next, I evaluated the model's QA accuracy on different problem types and found that it was less accurate on problems involving numerical computation. To this end, I introduce a numerical computation library: when dealing with QA that requires numerical computation, the model is first generated into a structured skeleton containing computation types and data, and the answers are then obtained through accurate numerical computation and fed back to the user.

In my future work, I will focus on the following two elements:

1. **Chart to Table** focuses on extracting structured data from charts for further analysis and utilization. This solves the problem that most of the datasets for current chart-related tasks are based on machine synthesis and are not adapted to real-world scenarios.

2. **Chart to Code** automatically generates the corresponding code given a chart and a specified code format. Figure 5 shows an example application of chart-to-code in a real scenario.

These tools aim to improve the utility and efficiency of processing chart data by combining visual representations with structured, manipulable formats.

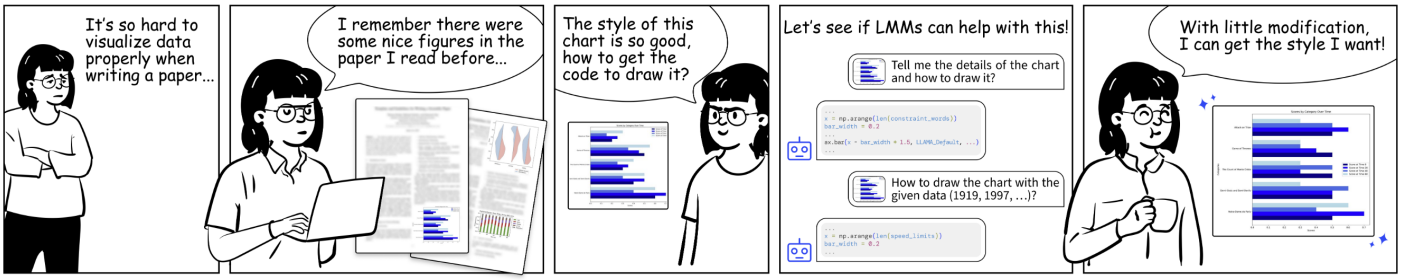


Figure 5. The real-world example. LMMs assist scientists and researchers in understanding, interpreting and creating charts during the reading and writing of academic papers. These models serve as assistants that enhance the comprehension and presentation of data in scholarly communications.

## References


[1] Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. ChartLlama: A Multimodal LLM for Chart Understanding and Generation, November 2023.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. Plotqa: Reasoning over scientific plots. *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1516–1525, 2019.

[4] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[5] Xingchen Zeng, Haichuan Lin, Yilin Ye, and Wei Zeng. Advancing Multimodal Large Language Models in Chart Question Answering with Visualization-Referenced Instruction Tuning, August 2024.

[6] Zhi-Qi Cheng, Qi Dai, and Alexander G Hauptmann. Chartreader: A unified framework for chart derendering and comprehension without heuristic rules. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22202–22213, 2023.

[7] Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. DePlot: One-shot visual language reasoning by plot-to-table translation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10381–10399, Toronto, Canada, July 2023. Association for Computational Linguistics.

[8] Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Martin Eisenschlos. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering. *arXiv preprint arXiv:2212.09662*, 2022.

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

[10] Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. ChartAssistant: A Universal Chart Multimodal Language Model via Chart-to-Table Pre-training and Multitask Instruction Tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7775–7803, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[11] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR, 2023.

[12] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang,

Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling, December 2024.

[13] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning, December 2023.

[14] Benny Tang, Angie Boggust, and Arvind Satyanarayan. VisText: A Benchmark for Semantically Rich Chart Captioning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7268–7298, Toronto, Canada, 2023. Association for Computational Linguistics.

[15] Gen Luo, Yiyi Zhou, Yuxin Zhang, Xiawu Zheng, Xiaoshuai Sun, and Rongrong Ji. Feast Your Eyes: Mixture-of-Resolution Adaptation for Multimodal Large Language Models, March 2024.

[16] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

[17] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[18] 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*, pages 8748–8763. PMLR, 2021.

[19] S Woo, S Debnath, R Hu, X Chen, Z Liu, IS Kweon, and S ConvNeXt Xie. V2: Co-designing and scaling convnets with masked autoencoders. arxiv 2023. *arXiv preprint arXiv:2301.00808*, 2023.

[20] Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. MMC: Advancing multimodal chart understanding with large-scale instruction tuning. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1287–1310, Mexico City, Mexico, June 2024. Association for Computational Linguistics.