ChartLlama: A Multimodal LLM for Chart Understanding and Generation

Anonymous ACL submission

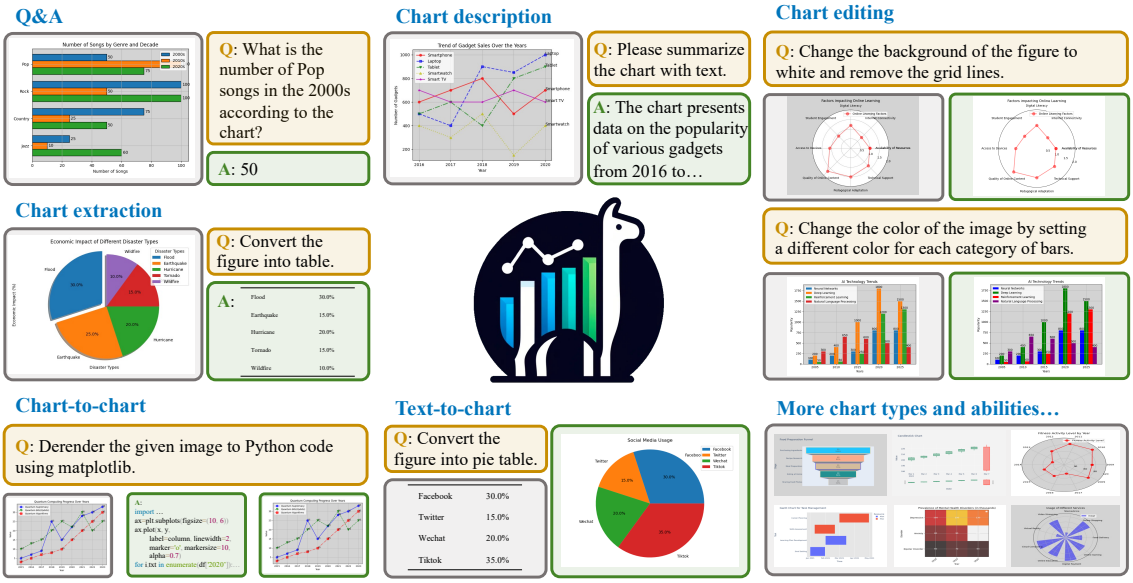


Figure 1: **Capability demonstration of ChartLlama.** An instruction-tuning dataset is created based on our proposed data generation pipeline. We train ChartLlama on this dataset and achieve the abilities shown in the figure.

Abstract

Multi-modal large language models have demonstrated impressive performances on most vision-language tasks. However, the model generally lacks the understanding capabilities for specific domain data, particularly when it comes to interpreting chart figures. This is mainly due to the lack of relevant multi-modal instruction tuning datasets. In this article, we create a high-quality instruction-tuning dataset leveraging GPT-4. We develop a multi-step data generation process in which different steps are responsible for generating tabular data, creating chart figures, and designing instruction tuning data separately. Our method's flexibility enables us to generate diverse, high-quality instruction-tuning data consistently and efficiently while maintaining a low resource expenditure. Additionally, it allows us to incorporate a wider variety of chart and task types not yet featured in existing datasets. Next, we introduce ChartLlama, a multi-modal large language model that we've trained using our created dataset. ChartLlama outperforms all prior methods in ChartQA, Chart-to-text, and Chart-extraction evaluation benchmarks. Additionally, ChartLlama significantly improves upon the baseline in our specially compiled chart dataset, which includes new chart and task types. The results of ChartLlama confirm the value and huge potential of our proposed data generation method in enhancing chart comprehension.

1 Introduction

In the past year, the field of artificial intelligence has undergone remarkable advancements. A key highlight is the emergence of large language models (LLMs) like GPT-4 (OpenAI, 2023). These models (Ouyang et al., 2022; Zeng et al., 2022; Team, 2023; Baichuan, 2023; Touvron et al., 2023a,b) have demonstrated a remarkable capability to comprehend and generate intricate textual data, opening doors to myriads of applications in both academia and industry. Taking this progress a step further, the introduction of GPT-4V (Yang

et al., 2023) marked another milestone. It endows LLMs with the ability to interpret visual information, essentially providing them with a vision. As a result, they can now extract and analyze data from images, marking a significant evolution in the capacities of these models.

However, despite the achievements and potentials of models like GPT-4V, the details behind GPT-4V's architecture remain a mystery. This opacity has given rise to questions within the academic world about the best practices for designing multi-modal LLMs. Notably, pioneering research initiatives, like LLaVA (Liu et al., 2023c,b) and MiniGPT (Zhu et al., 2023; Chen et al., 2023), provide insightful directions in this regard. Their findings suggest that by incorporating visual encoders into existing LLMs and then fine-tuning them using multi-modal instruction-tuning datasets, LLMs can be effectively transformed into multi-modal LLMs. It's noteworthy that these multi-modal datasets are typically derived from established benchmarks, presenting a cost-effective method for accumulating data required for instruction tuning.

Datasets grounded on established benchmarks, such as COCO (Lin et al., 2014), have significantly enhanced the abilities of multi-modal LLMs to interpret everyday photographs adeptly. However, when confronted with specialized visual representations, such as charts, they reveal a noticeable limitation (Yang et al., 2023; Liu et al., 2023a). Charts are important visual instruments that translate complex data sets into digestible visual narratives, playing a crucial role in facilitating understanding, shaping insights, and efficiently conveying information. Their pervasive presence, from academic publications to corporate presentations, underscores the essentiality of enhancing the capability of multi-modal LLMs in interpreting charts. Indeed, gathering data specifically to refine instructions for understanding charts presents several challenges. These typically stem from two areas: understanding and generation. An effective chart understanding model should be capable of extracting and summarizing data from various types of charts and making predictions based on this information.

However, most existing datasets (Masry et al., 2022; Kantharaj et al., 2022; Methani et al., 2020; Masry et al., 2023) only provide support for simple question-answering or captioning, primarily due to the absence of detailed chart information and annotations that provide a high-level understanding of raw data. The high dependency on manually annotated charts gathered by web crawlers negatively affects the quality of these datasets. Thus, the previous annotating methods could only result in chart datasets with lower quality and less comprehensive annotations. Compared with chart understanding, generating chart figures is a more challenging task for the model because existing deep-learning-based generation methods (Ramesh et al., 2021; Rombach et al., 2021) struggle to accurately create images based on instructions. Using Python code to generate charts seems promising which needs the corresponding annotations to supervise models. Most charts obtained from the web are devoid of detailed annotations, making it challenging to annotate the generation code. The absence of code annotations makes it challenging to supervise models in code generation. These issues combined impede the model's ability to understand charts and learn generation jointly.

To address this, we introduce an adaptive and innovative data collection approach exclusively tailored to chart understanding and generation. At the heart of our methodology is the strategic employment of GPT-4's robust linguistic and coding capabilities, which facilitate the creation of rich multi-modal datasets. This innovative integration not only optimizes data accuracy but also ensures its wide-ranging diversity. Specifically, our method comprises three main phases:

1) **Chart Data Generation**. Our strategy for data collection stands out for its flexibility. Rather than limiting data collection to conventional data sources such as the web or existing datasets, we harness the power of GPT-4 to produce synthesized data. By providing specific characteristics such as topics, distributions, and trends, we guide GPT-4 to produce data that is both diverse and precise.

2) **Chart Figure Generation**. Subsequently, GPT-4's commendable coding skills are utilized to script chart plots using the open-sourced library, like Matplotlib, given the data and function documentation. The result is a collection of meticulously rendered charts that span various forms, each accurately representing its underlying data.

3) **Instruction data generation**. Beyond chart rendering, GPT-4 is further employed to interpret and narrate chart content, ensuring a holistic understanding. It is prompted to construct relevant question-answer pairs correlating with the charts. This results in a comprehensive instruction-tuning corpus, amalgamating the narrative texts, question-answer pairs, and source or modified codes of the

charts.

A standout feature of our methodology is its flexibility, which diminishes the potential for bias while simultaneously offering scalability. Building on this robust methodology, we've crafted a benchmark dataset, which is made available for public access. This dataset stands out, not only for its superior quality but also its unparalleled diversity. To showcase the superiority of our benchmark, we introduced a multi-modal Large Language Model (LLM) named ChartLlama trained with our established benchmarks. Our extensive experiments evaluated on multiple existing benchmark datasets show that our model outperforms previous methods with remarkable advantages and considerably less training data. Additionally, ChartLlama is equipped with several unique capabilities, including the ability to support a wider range of chart types, infer across multiple charts, undertake chart de-rendering tasks, and even edit chart figures.

Our main contributions are summarized as follows:

- We introduce a novel multi-modal data collection approach specifically designed for chart understanding and generation. The proposed data collection method boasts superior flexibility and scalability, enabling easy migration to different types of charts and various tasks.
- Through our innovative data collection approach, we create a benchmark dataset that stands out in terms of both quality and diversity. We make this dataset publicly available to catalyze further advancements in the field.
- We develop ChartLlama, a multi-modal LLM that not only surpasses existing models on various existing benchmarks but also possesses a diverse range of unique chart understanding and generation capabilities.

2 Related work

2.1 Large Language Model

The series of LLM models, such as GPT-3.5 (Ouyang et al., 2022) and GPT-4 (OpenAI, 2023), have demonstrated remarkable reasoning and conversational capabilities, which have garnered widespread attention in the academic community. Following closely, a number of open-source LLM (Baichuan, 2023; Touvron et al., 2023a,b; Zeng et al., 2022; Bai et al., 2023a) models emerged, among which Llama (Touvron et al., 2023a) and Llama 2 (Touvron et al., 2023b) are notable representatives. With extensive pre-training on large-scale datasets and carefully designed instruction datasets, these models have also showcased similar understanding and conversational abilities.

2.2 Multi-modal Large Language Model

Concurrently, the academic community has witnessed a surge of development in multi-modal LLMs (Li et al., 2023a; Ye et al., 2023; Li et al., 2023c,b; Zhang et al., 2023b; Hu et al., 2023; Zhao et al., 2023; Bai et al., 2023b; Zhang et al., 2023a; Liu et al., 2023b,c; Chen et al., 2023; Zhu et al., 2023) built upon existing open-source models. With the exploration of training strategies and an increase in dataset scale, the performance of these new models has steadily improved, reaching comparable levels to GPT-4V in specific evaluation metrics. Notably, LLaVA-1.5 (Liu et al., 2023b), an iterative version of LLaVA, has gained popularity as a baseline due to its user-friendly training framework, superior performance, and data efficiency. Our work is also based on LLaVA-1.5.

2.3 Chart Understanding

In evaluations such as the report of GPT-4V (Yang et al., 2023) and HallusionBench (Liu et al., 2023a), it is evident that current multi-modal LLMs still struggle with complex chart-related problems. There are already some datasets (Methani et al., 2020; Kantharaj et al., 2022; Masry et al., 2022) available for evaluating models' chart understanding capabilities, mainly divided into two categories. One category measures through simple question-and-answer tasks, such as ChartQA (Masry et al., 2022), which has high-quality questions and answers annotated by humans. The other category converts charts into textual descriptions, with Chart-to-text (Kantharaj et al., 2022) being a representative work in this field. The charts and annotations in these datasets are derived from the real world, ensuring higher quality, and encouraging models to delve deeper into the trends and meanings behind the charts. Previous works focusing on chart understanding tasks can be divided into two main kinds of approaches. One kind of approach is using a single model to understand the charts and answer questions in natural language, for example, (Masry et al., 2023; Liu et al., 2022b). The other kind of approach, such as (Liu et al., 2022a; Xia et al., 2023), is to first utilize the model to convert the

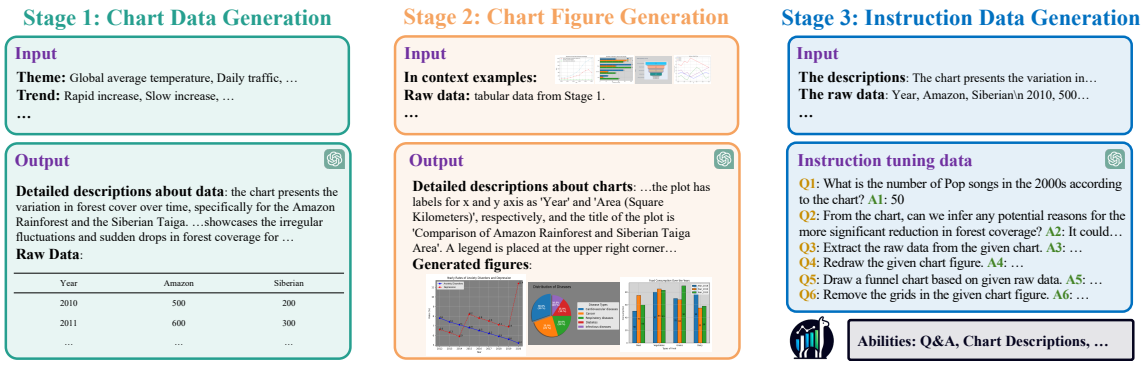


Figure 2: **Pipeline of our data generation method**. The innovative data generation process we proposed consists of three important steps relying on GPT-4. The dataset generated using this process exhibits significant advantages compared to previous datasets in terms of data diversity, quality, the number of chart types, and the variety of tasks. ChartLlama, which is trained on this dataset, has the ability to perform various tasks based on the design of the instruction-tuning data.

charts into structured data and then analyze and answer questions based on the structured data using existing large models. In our work, we primarily explore the former kind, aiming to leverage a single model to complete the entire process of chart understanding.

3 Method

In this section, we detail our unique approach to chart understanding and generation. Our method involves three interconnected steps: data collection, chart figure generation, and instruction data generation. We illustrate this process in Fig. 2. These steps are detailed in the following subsections.

3.1 Chart Data Generation

Our primary goal in chart data collection is to collect diverse and high-quality data. We employ two main strategies for this purpose: 1) Data Generation from Scratch Using GPT-4: To collect a diverse and high-quality dataset, we initially generate tabular data from scratch using GPT-4. We instruct GPT-4 to create data tables based on specific themes, distributions, and other characteristics like the size of the dataset in terms of rows and columns. This process ensured the creation of data with known and controlled characteristics, which can be essential for generating reliable instruction-answer pairs. Moreover, by managing these characteristics, we can intentionally minimize bias, leading to a more balanced dataset. 2) Synthesizing Data from Existing Chart Datasets. Our second strategy is to synthesize data by referencing existing chart datasets. These datasets already encompass a range of topics and characteristics, providing a solid base for data generation. By prompting GPT-4 with these datasets, we guide it to generate reasonable data that complements its existing knowledge base. This method added variety to our dataset and improved its overall quality.

Generating diverse data at scale using the LLM is not an easy task. When the prompt is designed improperly, the model tends to generate repetitive and meaningless data that deviates from the distribution of real-world data and thus lacks valuable insights that could be important for designing meaningful question-and-answer tasks. If we simply provide a set of data and require the model to imitate without any additional guidance, the model will probably just repeat the reference data. Therefore, in this step, it is necessary to provide the model with additional information, such as the topic and distribution, to ensure that it can be properly guided to generate meaningful data. We will now explain these pieces of information in detail.

Chart theme: We first generate hundreds of possible themes, which are all short phrases. When we generate data, we randomly select one from all those themes, which makes the data meaningful and diverse. This also makes it much more easy to generate questions and responses for instruction tuning.

Data trends: Another important characteristic of the data is the trends. We first generate several typical trend descriptions, like steadily increasing and suddenly dropping, then randomly select a few trends and require the model to generate data following them. If lacking such characteristics, the model will tend to generate several sets of data with meaningless distributions.

Column and row lengths: The lengths of columns and lengths are also necessary for data generation. Without specific constraints, LLMs tend to generate excessively long or even repetitive data, which is difficult to present in a meaningful way through charts.

Chart types: Charts of different types usually share different characteristics. For example, the sum of the values in pie charts should be 100%. If not specify the type of chart, we might end up generating data that doesn't comply with the corresponding chart standards.

3.2 Chart Figure Generation

The next step is to transform our dataset into visual charts using GPT-4's coding capabilities. We used popular chart plotting libraries, such as Matplotlib, as our primary tools. When prompting GPT-4, we provide the collected data, relevant function documentation, and in-context examples. We also give detailed instructions on diversifying aspects like color schemes and line types to enhance the visual appeal of the charts. To increase the diversity and success rate of our chart generation, we randomly sample successfully generated codes as in-context examples in the prompts. Compared with previous automated chart generation efforts that relied on templates, our approach offers greater variety and better visual appeal. It also enables us to generalize across different chart types effectively. The result was a collection of meticulously crafted charts, each accurately representing its data and visually appealing, showcasing the effectiveness of our method. The necessary input for the prompts in this stage is listed below.

Chart data: This is the most essential input for the task. The chart data is the information that will be visualized in the chart. Without it, no meaningful chart can be made.

Related function documentation: This is an important reference for generating the Python code. It provides information about the available functions and features that can be used to create the chart. With the documentation, the model could even create charts in new styles that are not in the in-context examples.

In context example: These in-context examples are sampled from pre-selected high-quality code. This helps to facilitate the construction of the Python code. When there is new generated code in high quality, we can save and sample it, which is used as in-context examples later.

Other requirements: To ensure that the final generated code is suitable for batch processing and execution, we also need to include several requirements in the prompt. For example, the data is required to be listed in the code to make the generated code self-contained and executable without the need for external files. We also set the requirements for the title, axis labels, legend, and text annotations. They provide context about what the chart represents and make it easier to understand the data. Without them, the chart can be confusing and difficult to interpret.

3.3 Instruction data generation

After completing the first two stages, we gathered comprehensive information about each chart, including precise tabular data, various characteristics from various perspectives, and the chart plotting code. Leveraging this rich information, we move on to generating a wide range of instruction-answer data with the assistance of GPT-4, significantly enhancing the capabilities of models trained on this dataset. In addition to fundamental chart understanding functionalities such as Q&A and summarization, our approach allows us to construct instructions and answers for more complex tasks, such as accurate data extraction, detailed chart descriptions, chart code generation, and even chart editing. Compared to previous pipelines for instruction data generation that often rely on human annotation, our methods yield significant time savings while enhancing diversity and quality in the resulting dataset.

Here are more details about the data that needs to be filled into the prompt.

Chart descriptions and raw data: providing these descriptions helps the model understand the context better. The first description helps the model to understand the nature of the data, and the second description assists in understanding the visual representation of the data. The raw data feeds the model with the actual values to base its responses on. All the descriptions and raw data are generated in the first and second stages.

Characteristics to be asked about: This requirement ensures that the model asks diverse and relevant questions about the chart. It prompts the model to explore different features of the data and its representation.

Method	Chartqa			Chart-to-text		Chart extraction (human)		Chart extraction (augmented)	
	Human	Augmented	Average	Pew	Statista	Precision	F1	precision	F1
Pix2struct (Lee et al., 2023)	30.50	81.60	56.00	10.30	38.00	–	–	–	–
Matcha (Liu et al., 2022b)	38.20	90.20	64.20	12.20	39.40	–	–	–	–
DePlot (Liu et al., 2022a)	–	–	–	–	–	81.32	81.15	93.42	93.29
Unichart (Masry et al., 2023)	43.92	88.56	66.24	12.48	38.21	61.51	35.20	79.59	70.21
Baseline* (Liu et al., 2023b)	37.68	72.96	55.32	7.16	24.65	53.48	48.39	55.17	49.50
ChartLlama	**48.96**	**90.36**	**69.66**	**14.23**	**40.71**	**84.92**	**84.89**	**94.94**	**94.78**

Table 1: **Results on traditional tasks.** We compare our work with the previous three open-source models and also compare it with Baseline* trained on the training split of respective benchmarks.

Method	Detailed Description	Chart-to-chart		Text-to-chart		Chart-editing		Chart-to-text	
		GPT Score	Success Rate (%)	GPT Score	Success Rate (%)	GPT Score	Success Rate (%)	Pew	Statista
LLaVA-1.5 (Liu et al., 2023b)	67.2	64.8	46	62.2	77	51.6	38	65.8	73.4
ChartLlama	**74.2**	**74.4**	**73**	**81.6**	**81**	**75.6**	**71**	**81.0**	**92.6**

Table 2: **Results on new tasks.** We primarily compared our work with the baseline model LLaVA-1.5. For the proposed new task, we used GPT for evaluation and validated the effectiveness of our proposed dataset. Evaluation of Chart-to-text using ChatGPT is also listed.

Chart type	Unichart	Baseline*	**ChartLlama**
Funnel	18.30	49.32	**70.59**
Gantt	9.80	40.17	**56.64**
Heatmap	25.43	38.18	**53.18**
Scatter	26.32	37.91	**54.97**
Box	16.67	28.33	**37.33**
Candlestick	15.79	25.69	**46.20**

Table 3: **Performances of Q&A on more categories of chart.** Baseline* means a modified version of LLaVA-1.5, which is further trained on the ChartQA dataset. We evaluate the performance of Baseline* and the previous state-of-the-art model Unichart on these new chart types.

	Human	ChatGPT	p-value	win-rate
LLaVA-1.5	66.78	62.32	1.32e-5	68%
ChartLlama	78.01	77.36	3.98e-7	32%

Table 4: **More evaluation metrics**. The evaluation scores of our evaluation metrics and human evaluation metrics are highly aligned and verify the effectiveness of our evaluation metrics.

4 Experiment

4.1 Implementation details

Implementation details. We train ChartLlama based on LLaVA-1.5 which provides fundamental abilities crucial for chart understanding and generation, including the OCR functionality. The projection layer and LLM are trained on our proposed dataset. Details of the model architecture and training hyper-parameters can be referred to in our appendix.

Dataset statistics. We show the statistics of our generated dataset in the Appendix. In our instruction-tuning data, Q&A dominates while the other tasks correspond to similar proportions of data. This is mainly because a single chart could be utilized to construct multiple Q&A data. Previous datasets usually gather only three types of charts: bar charts, line charts, and pie charts. Unlike them, we support a wide range of chart types. This is mainly due to the strong flexibility of our data construction method. It's worth noting that we can continue to expand on more data and chart types.

4.2 Evaluation Benchmark and Metrics

We evaluate possible models on seven tasks, including both the traditional tasks and novel tasks which verifies that our data generation pipeline has good scalability towards various tasks and chart types.

Traditional Tasks. Three traditional tasks are evaluated, namely ChartQA, Chart-to-text, and Chart-extraction. We use Relaxed Accuracy (Masry et al., 2022), GPT-4 as metrics for evaluation, and Precision and F1 scores as metrics, respectively. The details are listed in the Appendix.

New tasks. In addition to traditional tasks, we have devised four additional innovative tasks, three of which are targeted at chart generation to verify the scalability of novel tasks.

1) Detailed description. This task necessitates a comprehensive description of the given chart figure in a detailed manner, rather than summarizing it. The evaluation metric for detailed description is similar to the evaluation metric in Chart-to-text using GPT-4. We include detailed descriptions of the data and chart figures as conditions for GPT-4 to assist evaluation.

2) Chart-to-chart. This task aims to reconstruct the given chart figure. We design comprehensive evaluation metrics for code generation and utilize GPT-4 to measure the quality of the code. For the

chart-to-chart task, we evaluated the precision of data, axes, colors, chart types, and titles, rating from 0 to 5. Then we average them as the score for each sample. Finally, we normalize it to a range of 0 to 100 for easier analysis and report the average score across the entire test set.

3) Text-to-chart. The task aims at generating chart figures according to instructions and tabular data. We provide the input instructions and the generated code as conditions for evaluation criteria. The evaluation focuses mainly on visual similarity, completeness, accuracy, and aesthetics. Each standard is equally rated from 1 to 5 points. After averaging and normalization, we get the final score.

4) Chart-editing. The input condition for this task is a chart figure and an instruction describing how to edit the chart. It is expected to create a new figure that has been modified according to instructions based on the given chart figure. The evaluation method for chart-editing uses a similar process to previous chart generation-related tasks. The input conditions include the code of the chart to be modified, instructions, and the generated code of the model. The data accuracy, completeness, aesthetics, and instruction following performance are scored on a scale from 0 to 5. After averaging and normalization, the final result is obtained.

The alignment between our Evaluation Metrics and Human Evaluation. The results are listed in Table 4. We randomly select 100 samples for evaluation. Three volunteers were selected to evaluate the performance using the same criteria that were applied to GPT-4. Furthermore, we calculate the correlation between the scores given by humans and those given by ChatGPT to demonstrate that our metrics, which use ChatGPT as the evaluator, are reasonable. We also show the win-rate evaluation.

4.3 Results

We first compare our methods with existing chart understanding models, such as Pix2Struct (Lee et al., 2023), Matcha (Liu et al., 2022b), unichart (Masry et al., 2023). Then we further construct Baseline* using the same model architecture (Liu et al., 2023b) as ours, but is trained on the training split of each dataset separately. On traditional tasks, we have also tried to compare with existing multimodal large language models such as InternLM-XComposer (Zhang et al., 2023a), MiniGPT-v2 (Chen et al., 2023), and vanilla LLaVA (Liu et al., 2023b). However, we found the limitation of their instruction-following ability makes it hard to be evaluated by existing metrics.

ChartQA. ChartLlama achieves the best performance on both human and augmented splits of ChartQA (Masry et al., 2022) as listed in Table 1. Our ChartLlama also succeeds LLaVA-1.5 trained on the Unichart dataset, which is shown in the Appendix in detail. Previous methods typically involved pretraining on larger datasets and then finetuning on the training split of the same datasets to achieve better results, while ChartLlama does it in a zero-shot way after training on our dataset. Notably, although previous methods are trained on the ChartQA's training split, our method achieves significant advantages using much less data as shown in Table 7. Besides, we also evaluate our model on charts of novel types as shown in Table 3. Our model gains significant improvement towards Unichart and the Baseline*. This shows the superiority of ChartLlama in the ability to understand charts in novel charts.

Chart-to-text. As shown in Table 1 and Table 2, our method consistently outperforms the previous state-of-the-art approaches under different evaluation metrics and splits in Chart-to-text (Kantharaj et al., 2022). The improvement in our performance primarily stems from the model's ability to handle long texts. Previous works often encountered meaningless repetitions at the end of sentences when dealing with relatively longer texts.

Chart extraction. Our model performed the best in this task on ChartQA (Masry et al., 2022) as listed in Table 1. ChartLlama has been trained on a variety of instruction-tuning data, which greatly improved its ability to understand chart figures. This is the reason why it can significantly outperform LLaVA-1.5 in terms of performance.

Detailed description. ChartLlama gains significant performance improvement over LLaVA-1.5 which is shown in Table 2. The detailed description task requires the model's ability to understand image details, which can be significantly improved during the training for tasks related to chart figures.

Chart generation and modification. In Table 2, we compare our method with the original LLaVA-1.5, and we can see that our model gains consistent improvement over three tasks. LLaVA-1.5, which is the base model of ChartLlama, processes strong abilities to follow instructions and generate Python code, and thus also gains reasonable performances on chart generation and modification tasks.

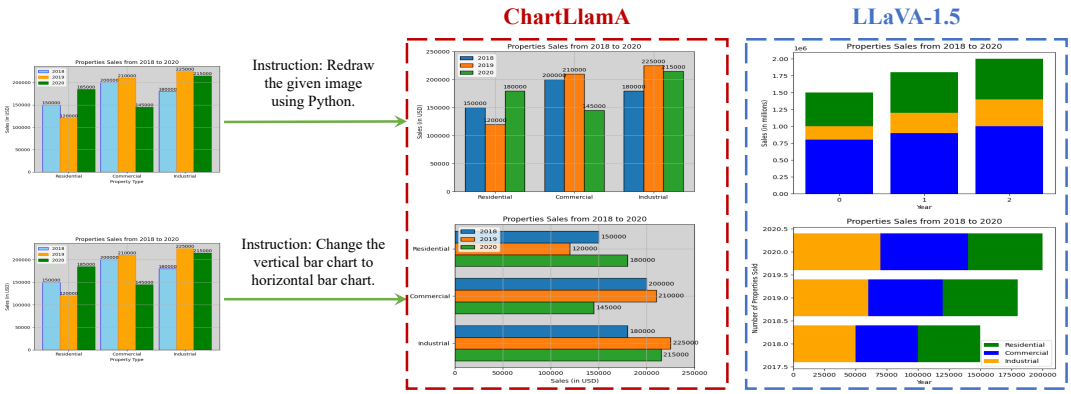


Figure 3: **Qualitative comparison for Chart-to-chart and Chart editing tasks.** We present the output results of LLaVA-1.5 and ChartLLaMA for the same chart given different instructions. The instruction in the first row requires the model to output the original chart, performing the chart-to-chart task. The instruction in the second row requires the model to output a horizontal bar chart, performing the chart editing task.

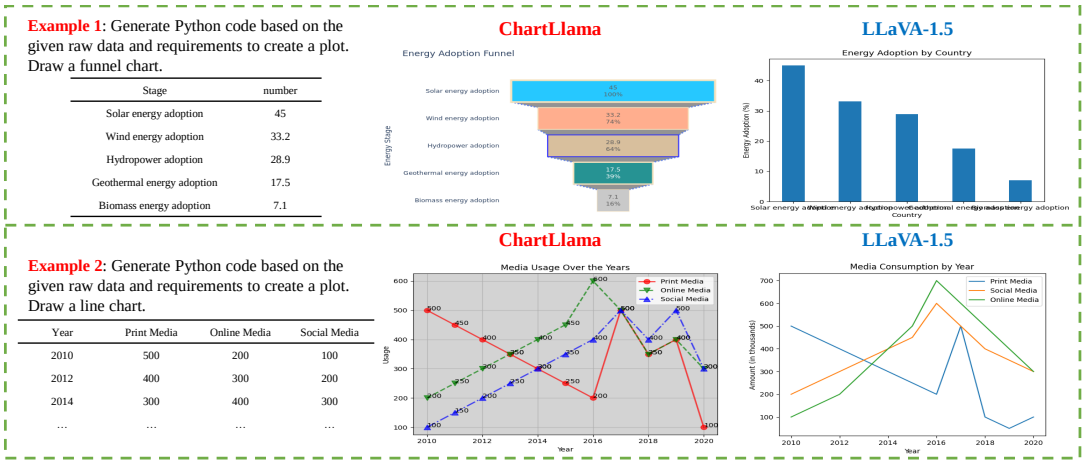


Figure 4: **Qualitative comparison for Text-to-chart task.** We have presented the generated images by ChartLLaMA and LLaVA-1.5 given the tabular data and the specified requirements.

4.4 Qualitative results

Figure 3 visualizes the chart-to-chart and chart-editing results. ChartLlama plots with the correct color and chart type, while LLaVA-1.5 cannot guarantee correctness. Figure 4 shows the text-to-chart results of ChartLlama and LLaVA-1.5. In the first example, ChartLlama successfully generates a funnel chart following the instructions and plots correct values. But LLaVA-1.5 even cannot draw funnel charts. In the second example, it is obvious that the result of ChartLlama contains more details and adds data values for human convenience. Both two examples show the strong ability of chart-generating and editing abilities of ChartLlama.

5 Conclusion

In this paper, we propose a flexible and robust approach for synthesizing instruction-tuning data for chart data. Then we train ChartLlama on the proposed dataset. The data generation flow we propose greatly reduces the difficulty of generating chart-related data and improves the controllability and diversity of the generated data. Experiments conducted on both traditional tasks and our newly constructed tasks validate the outstanding performance of ChartLlama. Thanks to the diverse instruction-tuning data in our dataset, ChartLlama possesses various capabilities that were absent in previous models. Moreover, its ability to comprehend both instructions and figures can easily extend to new categories of chart figures or tasks. We believe that our data generation process can make significant contributions to multimodal LLM in tasks related to chart understanding. Furthermore, it will facilitate the application of similar data generation processes in other domains.

6 Limitations

The current version of ChartLlama's vision encoder lacks the ability to handle multilingual OCR tasks, restricting the model's utility for charts containing non-English text. To overcome this limitation, we are contemplating the creation of a novel vision encoder that boasts proficiency in multilingual OCR tasks.

Ethics Statement

This section provides a comprehensive reflection on the broader impacts and ethical considerations associated with our multi-modal language model, ChartLlama. ChartLlama, designed to improve machine understanding of chart figures by using a high-quality instruction-tuning dataset, prompts significant ethical contemplation, especially in terms of data privacy, data representation, accuracy, accessibility, and potential misuse.

Data Privacy: ChartLlama's design is founded on the principle of respecting user data privacy. The model interacts with data in a manner that does not require access to sensitive information, thereby ensuring user data confidentiality. We are committed to continually improving these privacy measures to protect user data.

Data Representation and Accuracy: The data used in generating the instruction-tuning dataset is carefully selected to ensure diversity and eliminate bias. Moreover, we recognize the vital importance of accuracy in our model's interpretation of chart figures. We are dedicated to continually refining our model to enhance its interpretative accuracy and reliability.

Accessibility: Our model aims to make chart figures more accessible and understandable to users who may find them challenging to interpret, including individuals with visual impairments and those not familiar with data visualization techniques. We acknowledge the current limitations of our model in this regard and are committed to improving its capabilities to make it more inclusive.

Sustainability: Aware of the environmental impact associated with training large AI models, our proposed data generation method aims to generate high-quality tuning data efficiently while maintaining low resource expenditure. This approach is part of our commitment to minimizing the environmental footprint of our research and development activities.

Potential Misuse: An important ethical concern is the potential misuse of ChartLlama for malicious purposes, such as misrepresenting data or manipulating chart figures. We strictly oppose the use of our model for any unethical practices. To combat this, we advocate for:

- **Monitoring and Detection**: Implementing tools to detect the misuse of ChartLlama, particularly in data misrepresentation.
- **Ethical Guidelines and Governance**: Establishing and enforcing ethical guidelines for the use of ChartLlama, ensuring researchers and developers are aware of the ethical implications of their work.
- **Collaboration with Stakeholders**: Working with data scientists, researchers, and users to align ChartLlama's use with ethical standards and best practices in data visualization and interpretation.

Through this ethics statement, we aim to underscore our dedication to responsible innovation, emphasizing the importance of safeguarding against potential misuse of technology while promoting the beneficial potentials of AI in chart comprehension and data visualization.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023a. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023b. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

Baichuan. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechu Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed

Elhoseiny. 2023. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Wenbo Hu, Yifan Xu, Yi Li, Weiyue Li, Zeyuan Chen, and Zhuowen Tu. 2023. Bliva: A simple multimodal llm for better handling of text-rich visual questions.

Shankar Kantharaj, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022. Chart-to-text: A large-scale benchmark for chart summarization.

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

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. 2023a. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*.

Juncheng Li, Kaihang Pan, Zhiqi Ge, Minghe Gao, Hanwang Zhang, Wei Ji, Wenqiao Zhang, Tat-Seng Chua, Siliang Tang, and Yueting Zhuang. 2023b. Fine-tuning multimodal llms to follow zero-shot demonstrative instructions. *arXiv preprint arXiv:2308.04152*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023c. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2022a. Deplot: One-shot visual language reasoning by plot-to-table translation. *arXiv preprint arXiv:2212.10505*.

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

Fuxiao Liu, Tianrui Guan, Zongxia Li, Lichang Chen, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. 2023a. Hallusionbench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v (ision), llava-1.5, and other multi-modality models. *arXiv preprint arXiv:2310.14566*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023b. Improved baselines with visual instruction tuning.

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

Yadong Lu, Chunyuan Li, Haotian Liu, Jianwei Yang, Jianfeng Gao, and Yelong Shen. 2023. An empirical study of scaling instruct-tuned large multimodal models.

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. *arXiv preprint arXiv:2305.14761*.

Ahmed Masry, Do Long, 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.

Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. 2020. Plotqa: Reasoning over scientific plots. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*.

R OpenAI. 2023. Gpt-4 technical report. arxiv 2303.08774. *View in Article*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

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.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, M Chen, R Child, V Misra, P Mishkin, G Krueger, S Agarwal, et al. 2021. Dall· e: Creating images from text. *OpenAI blog. https://openai.com/blog/dall-e*.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-resolution image synthesis with latent diffusion models.

InternLM Team. 2023. Internlm: A multilingual language model with progressively enhanced capabilities. https://github.com/InternLM/InternLM.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Renqiu Xia, Bo Zhang, Haoyang Peng, Ning Liao, Peng Ye, Botian Shi, Junchi Yan, and Yu Qiao. 2023. Structchart: Perception, structuring, reasoning for visual chart understanding. *arXiv preprint arXiv:2309.11268*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chaoya Jiang, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023. mplug-owl: Modularization empowers large language models with multimodality.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Wenwei Zhang, Hang Yan, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. 2023a. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition.

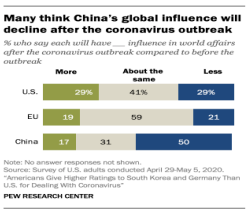
Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. 2023b. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*.

Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. 2023. Mmicl: Empowering vision-language model with multi-modal in-context learning. *arXiv preprint arXiv:2309.07915*.

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*.

Method	ChartQA			ChartQA on special charts					
	Human	Augmented	Average	Funnel	Gantt	Heatmap	Scatter	Box	Candlestick
Unichart (Masry et al., 2023)	43.92	88.56	66.24	18.30	9.80	25.43	26.32	16.67	15.79
InternLM-XComposer-VL (Zhang et al., 2023a)	8.48	7.36	7.92	12.42	6.36	16.18	18.13	15.33	16.96
Mini-GPT-v2 (Chen et al., 2023)	15.60	8.40	12.00	26.7	15.03	28.32	28.65	21.33	17.54
Qwen-VL (Bai et al., 2023b)	37.60	63.76	50.68	6.54	9.83	13.29	7.02	8.00	1.75
mPLUG-Owl2 (Ye et al., 2023)	21.20	22.0	21.60	23.53	27.75	19.08	16.37	15.33	19.30
Baseline* (Liu et al., 2023b)	37.68	72.96	55.32	49.32	40.17	38.18	37.91	28.33	25.69
ChartLlama	**48.96**	**90.36**	**69.66**	**70.59**	**56.64**	**53.18**	**54.97**	**37.33**	**46.20**

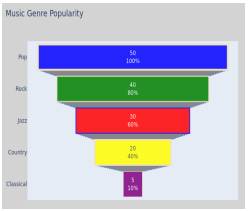
Table 5: **Results on traditional tasks.** We compare our work with the previous three open-source models and also compare it with Baseline* trained on the training split of respective benchmarks.



Example 1

Question: What's the average of all the values in the green bars (round to one decimal)?

Answer:
Unichart: 43
LLaVA-1.5: 38.5
ChartLLaMA: 21.6



Example 2

Question: What is the percentage decrease in popularity from the Country genre to the Classical genre?

Answer:
Unichart: 10
LLaVA-1.5: 50%
ChartLLaMA: 75%

Figure 5: **Visualization on the ChartQA task.** Here are two examples of the predictions of Unichart, LLaVA-1.5, and ChartLlama. Our proposed ChartLlama could follow the long instructions and do calculations to get the correct results.

Prompt Design	Successful Rate
Original	85%
w/o In context example	43%
w/o Documentation	65%
w/o Both	28%

Table 6: **Ablations on Prompt of Stage Two.** The first row shows the successful rate of our proposed data generation method in the second stage. Then we evaluate the generated results when removing the in-context examples, the documentation, and both of them, respectively.

Datasets	#Chart type	#Chart figure	#Instruction tuning data	#Task type
ChartQA (Masry et al., 2022)	3	21.9K	32.7K	1
PlotQA (Methani et al., 2020)	3	224K	28M	1
Chart-to-text (Kantharaj et al., 2022)	6	44K	44K	1
Unichart (Masry et al., 2023)	3	627K	7M	3
StructChart (Xia et al., 2023)	3	9K	9K	1
ChartLlama	**10**	**11K**	**160K**	**7**

Table 7: **Dataset statistics**. Thanks to the flexibility of our data construction method, our proposed dataset supports a wider range of chart types and tasks. We can generate more diverse instruction-tuning data based on specific requirements.

Appendix

A Model architecture

To elucidate our training strategies, we provide some clarification about the modifications in LLaVA-1.5 (Liu et al., 2023b), and introduce its essential model architectures.

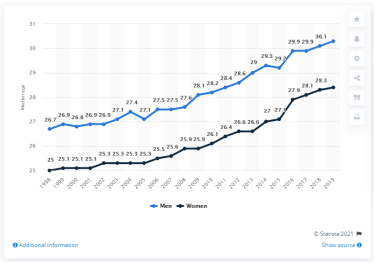
Vision encoder: LLaVA-1.5 incorporates CLIP's vision encoder (Radford et al., 2021). The primary distinction is that LLaVA-1.5 employs ViT-L/14@336px, while LLaVA uses ViT-L/14@224px. Another notable alteration concerns the image processor. Eschewing traditional center cropping, LLaVA-1.5 adopts padding as an image pre-processing technique, ensuring that all information in the provided image can be apprehended.

Projection layer: In LLaVA-1.5, the initial single linear layer is substituted with a two-layer MLP, resulting in improved performance.

Lora Layer: Based on experiments in (Lu et al., 2023; Liu et al., 2023b), implementing Lora (Hu et al., 2022) layers is sufficient to achieve performance comparable to full fine-tuning strategies. For the original LLaVA (Liu et al., 2023c), Lora layers with a Lora rank of 64 suffice, whereas for LLaVA-1.5 (Liu et al., 2023b), the Lora rank needs to exceed 128.

B Evaluation Metrics for Traditional Tasks

1) For ChartQA (Masry et al., 2022), we evaluate relaxed accuracy on human and augmentation split, respectively.
2) Chart-to-text contains two separate datasets for training and evaluation. BLEU-4 and GPT-4 serve as metrics for evaluation. BLEU-4 is widely used in many NLP tasks. However, the Chart-to-text datasets contain too few ground-truth references, which harms the evaluation metrics. To facilitate more reasonable evaluations, we propose a new evaluation metric based on GPT-4, referring to the GPTScore (Fu et al., 2023). We designed scoring criteria that require the ground-truth reference and



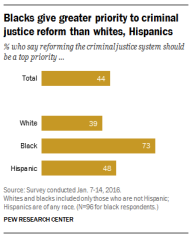
Question: Decompose the chart figure and present the base data.

Answer:

Unichart	2019	2018	2017	2016	2015	2014	2013	2012	2011	2010	2011	2002	25.0	25.0	25.0	25.0	...
Men	30.28	30.08	29.89	29.89	29.2	29.3	29.0	28.6	28.6	28.0	28.0	28.0	25.0	25.0	25.0	25.0	...
Women	28.39	28.3	28.11	27.9	27.11	27.0	26.6	26.6	26.6	26.6	26.4	26.0	25.0	25.0	25.0	25.0	...

ChartLLaMA	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	...
Men	26.7	26.9	26.8	26.5	26.7	26.9	27.1	27.1	27.6	28.1	28.3	28.3	28.6	29.0	29.4	29.9	...
Women	25.0	25.1	25.1	24.8	24.4	24.2	24.0	24.1	24.6	25.1	25.6	25.8	26.3	26.9	27.5	28.1	...

Figure 6: **Visualization of Chart-extraction.** We find that ChartLlama is especially good at long text processing. While the previous SOTA, Unichart, will generate meaningless redundant words when the output is too long.



Unichart: According to a survey conducted by the Pew Research Center, blacks give greater priority to criminal justice reform than whites. Specifically, 73% of blacks prioritize criminal justice reform as a top priority, while only **44% of whites** prioritize it. Hispanics prioritize criminal justice reform as a top priority, while only 48% of Hispanics prioritize it as a priority. **The chart shows that the majority of blacks. The chart reveals a top priority, with only 39% of them as a top priority, where Hispanics, where Hispanics priorities, where Hispanics priorities, where Hispanics priorities priorities priorities priorities priorities priorities……**

LLaVA-1.5: The chart shows the percentage of people who believe that blacks, whites, and Hispanics should have different priorities in the criminal justice system. The results are broken down by race, **with 43% of whites**, 73% of blacks, and **47% of Hispanics** believing that blacks should have a higher priority. **The chart also displays the total number of people surveyed for each race, with a total of 1,000 people participating in the survey.**

ChartLLaMA: The data indicates a significant divergence in the perception of prioritizing the reform of the criminal justice system among different racial groups. A striking 73% of blacks express the view that this should be a top priority, contrasting with 48% of Hispanics and merely 39% of whites who share the same sentiment.

Figure 7: **Visualization of Chart-to-text.** We select one image from the Pew Dataset and show the results of Unichart, LLaVA-1.5, and ChartLlama. We find that Unichart easily falls into repeated words again and LLaVA-1.5 suffers from hallucination.

raw data as input conditions. Details can be found in the appendix.

3) Chart extraction aims to extract the tabular data from the given chart figure. We follow the evaluation framework of DePlot (Liu et al., 2022a) and report the Precision and F1 scores on the challenging ChartQA dataset, which also provides the tabular data for each chart figure.

C Dataset Scale and Statistics

Human Evaluation of Data Quality. We sample 100 instances from our proposed dataset to assess the quality and then establish four criteria for volunteers to rate the quality of the provided training samples. These criteria gauge the aesthetics of the generated figure, the presence of chart occlusion, the accuracy of the given answer, and the quality of the image design. Each scoring criterion carries a maximum of one point. The average score for the provided images is 3.7/4.0. We also evaluate UniChart using the same criteria, and its average score is 3.1/4.0. This indicates the superior quality of our generated chart dataset.

The dataset's statistics are shown in Table 7 and Fig. 16. The model's training process is broken down into two critical stages: pretraining and fine-tuning. The primary objective of pretraining is to effectively initialize the vision projector while fine-tuning steers the Language Learning Model (LLM) to adhere to the provided instructions.

In the pretraining phase, LLaVA-1.5 utilizes approximately 558k image-caption pairs to train the projection layer. It is anticipated that the vision features will align with the language features to a certain extent. This dataset originates from a subset of around 558K image-text pairs from LAION-CC-SBU, each paired with a BLIP caption.

The fine-tuning phase involves further training of the model on 665k instruction-following data pairs. LLaVA-1.5 manifests an array of capabilities during this stage. The instruction-following data pairs are meticulously generated to encompass the required abilities. To enhance the model's capacities in varied contexts, additional academic-task-focused Visual Question-Answering (VQA) datasets for VQA, Optical Character Recognition (OCR), and region-level perception are incorporated. The final compilation includes several datasets: OpenKnowledge VQA (OKVQA, A-OKVQA), Region-level VQA (Visual Genome, RefCOCO), and OCR (OCRVQA, TextCaps). A-OKVQA is transformed into multiple-choice questions, employing a specific response formatting prompt: answer by directly specifying the option's letter from the provided choices.

ChatGPT for Chart Data Generation

You are an expert at generating data in csv format. You receive several key characteristics about the data. Your final output should include data in CSV format for the chart, and a comprehensive description of the chart data and figure.
##Expected characteristics of the data in the chart.
The theme of the chart is **[Selected chart theme]**. Different series of data in the chart can have different trends. The trends in the chart data should include as many of the given trends as possible: **[A list of possible trends]**.
The data should be **diverse and contain several outliers**. The numbers of rows and columns are **[Random number]** and **[Random number]**.
You can list the nouns you know, which are related with the theme, along the first column and row of the table.
Requirement about the description
The description should focus on several key elements: the chart's theme, the general trend of the data, individual trends within the data, the comparison between data, and any outliers present in the chart.
Requirement about the output
Your output should comprise the generated data wrapped in <data start> and <data end>, and detailed descriptions about the chart wrapped in <description start> and <description end>.

Figure 8: **The prompt template for Stage One.** This template is used for Chart Data Generation. Utilizing this template could guide GPT-4 to generate diverse raw tabular data and detailed descriptions of the content.

ChatGPT for Chart Figure Plotting

You are a specialist in two aspects, drawing charts with matplotlib or plotly, and providing detailed descriptions about the chart. You receive the data in the format of csv table. In addition, you are provided with an example of Python code drawing a chart for reference. You also receive some parameters that could be used to increase the diversity. You need to generate Python code to plot the given data as a chart figure and providing detailed description about the figure.
Additional requirements:
The chart should have the title, labels on x-axis and y-axis. The chart should have legend. You can annotate data values above the point on the chart figure. Do not use show function to show the figure. The csv data should be listed in the code.
The output contains two parts. The first part is the generated Python code wrapped in <code start> and <code end>.
Next is the detailed description about the chart wrapped in <description start> and <description end>.
The code should be able to be executed without external files.
The given data: **[Chart data]**.
The given code example: **[Code example]**.
As for additional parameters, you could consider: **[Related function documentation]**.

Figure 9: **The prompt template for Chart Figure Plotting.** Following such instructions, GPT-4 could generate codes that could draw chart figures using Python packages.

D Generation Prompt for ChartLlama

As listed in Figure 8, Figure 9, and Figure 10, we have provided standard prompts for data generation in three stages. The text in black color in the figure denotes the fixed prompt template, while the text in red color brackets requires filling in, which serves to enhance the diversity and controllability of the generated results. The detailed meanings of the different variables have already been discussed in the main text, thus we will not elaborate further.

E Ablation Study on the Conditions of Generation Prompt for ChartLlama

In order to verify the impact of our proposed generation process on the results, we designed an ablation experiment on the prompt for the second step, diagram construction, which is shown in Table 6. Specifically, we removed the in-context examples and the description of the function, then retested the probability of successful generation. The results show that combining both in-context examples and documentation could significantly improve the successful rate of plotting figures. Also, we observe that the diversity could also improve a lot, which is hard to quantify.

F Filtering Mechanism

The data generation process may produce some erroneous samples, but filtering and correcting these samples can be challenging because the samples contain figures that cannot be processed by GPT-4. We only performed basic error correction, in-

ChatGPT for Instruction Data Generation

You are an AI visual assistant that can analyze chart figures. You receive two detailed descriptions and raw data about the same chart. The first description is the information about the raw data in the chart. The second description is about the chart figure based on Python code. In addition, raw data values within the chart is given. Answer all questions as you are seeing the chart figure. Design a question-answer pair between you and a person asking about this chart figure. The answers should be a single word or phrase, and in a tone that a visual AI assistant is seeing the chart figure and answering the question.
Ask diverse questions and give corresponding answers. Include questions asking about **[Characteristics]** and so on.
Only include questions that have definite answers:(1) one can see in the chart figure that the question asks about and can answer confidently;(2) one can determine confidently from the chart figure that it is not in the chart figure.
Do not ask any question that cannot be answered confidently. The answers should be a single word or phrase.
Here are some examples and remember to follow their format: **[In context examples]**.
The first description: **[Description about chart data]**.
The second description: **[Description about chart figure]**.
The raw data: **[Raw data]**.

Figure 10: **The prompt template for Instruction Data Generation.** This step is targeted at generating various training data. To guarantee the quality and diversity of the generated samples, we need to give enough information on the chart figure and in-context examples.

cluding checking the data generation format and verifying the correct execution of the code. The data generation format check involves confirming whether the model has separated different data results with different markers according to our requirements. The check for correct code execution involves running the generated plotting script. If this script fails to run, we no longer use the training sample corresponding to that plot. Such basic data screening is sufficient to ensure the quality of the generated dataset. We are also considering incorporating more effective automatic screening mechanisms to avoid contamination of the dataset by poor-quality samples.

G Evaluation Prompt for ChartLlama

We have prepared five evaluation prompts in total, each tailored for a specific task: chart-to-text in Figure 15, detailed description in Figure 11, chart-to-chart in Figure 12, text-to-chart in Figure 13, and chart-editing in Figure 14. We have designed distinctive scoring criteria for different tasks and provided reference information based on the additional annotations in the dataset. Ultimately, we employed GPT-4 for scoring purposes.

H Comparison with Multi-modal LLMs

Comparison with LLaVA-1.5 Trained on Alternate Datasets. Of all the pre-existing chart-comprehending datasets, UniChart is the largest and encompasses the most task types. Thus, we compare our ChartLlama with a novel model trained on UniChart using the LLaVA-1.5 framework. The results pertinent to ChartQA-related tasks are presented in the table above. The evaluation metrics of ChartQA and Chart Extraction are Relaxed Accuracy and F1 score, respectively. As expected, the performance of LLaVA-1.5 trained on UniChart is subpar, because each sample only includes simple question-answer pairs or succinct captions, which fail to provide sufficient information for the model to interpret the provided chart figures.

Dataset	ChartQA		Chart Extraction	
	human	augmented	human	augmented
UniChart	29.36	40.0	26.54	60.40
Ours	48.96	90.36	84.89	94.78

Table 8: Comparison with LLaVA-1.5 trained on the dataset proposed in Unichart.

Traditional Tasks: The Table 5 includes a comparison of existing state-of-the-art (SOTA) models, illustrating their respective performances. Interestingly, some models (Zhang et al., 2023a) show unexpectedly low performance. This outcome is not a consequence of our experimental configuration. Rather, it derives from the fact that these models have not been trained on corresponding instruction-following tasks, which results in outputs that are incompatible with the evaluation framework. We argue that training these models specifically on instruction-following tasks using specific datasets would likely yield improved performance. Another notable observation is the performance gap

ChatGPT for Evaluation of Detailed Description

You are an expert tasked with evaluating the descriptions generated by a model. I will provide you with the ground-truth description and raw data, as shown below:
[Ground truth description]
[Raw data]
The description generated by the model is as follows:
[Predicted description]
Please refer to the above content and score the model on the given criteria. The criteria are as follows:
0 points:
- The model's description doesn't refer to the chart at all, or is completely irrelevant.
- It doesn't show any understanding of the chart figures or raw data.
1 point:
- The model's description refers to the chart but the details are largely incorrect.
- There is minimal understanding of the chart figures and raw data.
2 points:
- The model's description refers to the chart and some details are correct, but key elements are missing or incorrect.
- There is a basic understanding of the chart figures and raw data, but significant errors are present.
3 points:
- The model's description accurately refers to the chart and most details are correct.
- The model shows a good understanding of the chart figures and raw data, but there are some minor errors or omissions.
4 points:
- The model's description accurately refers to the chart and all details are correct.
- The model shows a very good understanding of the chart figures and raw data.
- There might be minor improvements possible in the description's clarity or completeness.
5 points:
- The model's description accurately and comprehensively describes the chart.
- The model shows an excellent understanding of the chart figures and raw data, with no errors or omissions.
- The description is clear, detailed, and precise. It could be used as a standalone explanation of the chart.
First return a single value (from 0 to 5) in the first line, then reply with your reason in the second line.

Figure 11: **The prompt template used for evaluation on the Detailed-description task.** The input conditions are the ground-truth description, raw data, and predicted description. GPT-4 will follow the given criteria and generate the final score and reasons.

of Qwen-VL between the ChartQA test splits and the ChartQA on our specially generated charts. Despite being trained on ChartQA, Qwen-VL underperforms on the specially generated charts, underscoring the effectiveness and need for our proposed benchmark. However, the lack of general training scripts provided by many models poses a challenge to our fine-tuning efforts. Nonetheless, our hypothesis finds support in the model LLaVA-1.5. Initially, LLaVA-1.5 performed poorly on the dataset but showed significant improvement when trained on the designated dataset.

Novel Tasks: We also conducted tests on the newly proposed tasks. However, most of the given dataset cannot generate executable Python code except LLaVA-1.5 (Liu et al., 2023b). We speculate that this is because these large multimodal models have been overtrained on visual language datasets, resulting in the loss of their code generation capabilities in Language Learning Models (LLMs); while LLaVA-1.5 adopted a series of optimization measures during its training process. For instance, compared to other large multimodal language models, LLaVA-1.5 has a shorter training time, fewer training parameters, a more moderate dataset scale, and incorporates pure text data during training to maintain the basic capabilities of LLMs. This experiment also suggests that if we expect the model to have a certain level of generalization ability, we should avoid making excessive adjustments to the LLMs. This is also why our ChartLlama model chose to train with fewer parameters.

I More Qualitative Results

ChartQA. As shown in Figure 5, we compare our ChartLLaMA with Unichart and LLaVA-1.5. The given examples are both related to longer questions and calculations, which is hard for Unichart. What's more, without the language understanding ability, Unichart even cannot follow complex instructions. In Example 2, the answer of Unichart is even not a percentage. Although LLaVA-1.5 has the ability of OCR and instruction-tuning, it cannot

identify which part of the image is related to the question because it has not been trained on chart figures. Thus, it fails in both examples, either.

Chart Extraction. As depicted in Figure 6, ChartLLaMA also possesses the capability to convert charts into structured data. Both the output results of Unichart and ChartLLaMA are a string of characters and we visualize it as tables for convenience. The first mistake of Unichart is reversing the order of years. Another mistake in Unichart is the persistent output of repetitive and meaningless characters at the end. Meanwhile, our proposed model, ChartLLaMA, benefits from strong language comprehension and output capabilities, which prevent the occurrence of such errors.

Chart Description. In Figure 7, we visualize the results of Unichart, LLaVA-1.5, and ChartLLaMA on the Chart-to-text task. The results from Unichart contain incorrect values and meaningless repetitions when generating long texts. LLaVA-1.5 performs better for long output sequences but suffers from wrong OCR recognition results and hallucinations. Our proposed ChartLLaMA performs best among these three models.

ChatGPT for Evaluation of Chart-to-chart

You are an expert tasked with evaluating the Python code generated by a model. I will provide you with the ground-truth code used for generating the chart figure, and the predicted python code. You need to evaluate the predicted Python code and score it from 0 points to 5 points. **Here is the criteria:**

0 points:
- The model's generated Python code either does not produce a chart at all, or the chart is entirely unrelated to the original.
- It doesn't show any understanding of the chart figure or the Python code used to produce it.

1 point:
- The model's generated Python code refers to the original chart but the details are largely incorrect.
- There is minimal understanding of the chart figure and the Python code used to produce it.

2 points:
- The model's generated Python code refers to the original chart and some details are correct, but key elements are missing or incorrect.
- There is a basic understanding of the chart figure and the Python code used to produce it, but significant errors are present.

3 points:
- The model's generated Python code accurately refers to the original chart and most details are correct.
- The model shows a good understanding of the chart figure and the Python code used to produce it, but there are some minor errors or omissions.

4 points:
- The model's generated Python code accurately refers to the original chart and all details are correct.
- The model shows a very good understanding of the chart figure and the Python code used to produce it.
- There might be minor improvements possible in the code's clarity or completeness.

5 points:
- The model's generated Python code accurately and comprehensively reproduces the original chart.
- The model shows an excellent understanding of the chart figure and the Python code used to produce it, with no errors or omissions.
- The generated Python code is clear, detailed, and precise. It could be used as a standalone code to draw the chart.

The ground-truth Python code:
[The Ground-truth code]

The predicted Python code:
[The predicted code]

First return a single value (from 0 to 5) in the first line, then reply with your reason in the second line.

Figure 12: **The prompt template used for evaluation on the Chart-to-chart task.** The input conditions are ground-truth code and predicted code. Following the given criteria, GPT-4 generates the score and corresponding reason.

ChatGPT for Evaluation of Text-to-chart

You are an expert tasked with evaluating the Python code generated by an LLM. The LLM could generate Python code for chart figure based on input raw csv data and instructions. I will provide you with the input raw data and the instruction. Also, I will give you a reference code, and the predicted python code by the LLM. You need to evaluate the predicted Python code and score it from 0 points to 5 points. **Here is the criteria:**

1. ****Correctness**:** This metric evaluates whether the generated code accurately fulfills the given instructions. The score could be binary (1 for correct, 0 for incorrect) or based on a proportion of test cases passed.
   Scoring Standard:
   - Score 5: The code fulfills all tasks perfectly.
   - Score 3: The code fulfills some tasks but has minor errors.
   - Score 0: The code does not fulfill the tasks or is entirely incorrect.
2. ****Readability**:** This metric assesses whether the code is easy to read and understand, which includes appropriate use of comments, variable names, and code structure.
   Scoring Standard:
   - Score 5: The code is very readable with good structure, comments, and variable names.
   - Score 3: The code is somewhat readable but could be improved.
   - Score 0: The code is not readable or poorly structured.
3. ****Visualization Aesthetics and Detailing**:** This metric evaluates the level of detailing in the generated figures and the aesthetics of the visualization. It assesses how well the code incorporates elements like color, labels, annotations, and other features to improve the look and interpretability of the graphs.
   Scoring Standard:
   - Score 5: The code consistently generates figures with excellent detailing and aesthetics. Graphs have appropriate and diverse color schemes, clear labels, and annotations, making them easy to interpret and visually appealing.
   - Score 3: The code generates figures with adequate detailing and aesthetics. Some elements like color, labels, or annotations could be improved for better interpretability and visual appeal.
   - Score 0: The code does not generate figures, or the figures generated lack any form of detailing or aesthetics, making them uninterpretable and visually unappealing.

The raw data and instruction:
[Raw tabular data and Instructions]

The reference Python code:
[Reference code]

The predicted Python code:
[Predicted code]

The output should first give the average score based on three criteria, then output scores for each criteria. The output should follow this format:
Average: number
Correctness: number
Readability: number
Visualization Aesthetics and Detailing: number

Figure 13: **The prompt template used for evaluation on the Text-to-chart task.** The input conditions are the raw tabular data and instructions, the reference code, and the predicted code. Finally, GPT-4 will return with the average score and scores for each criterion.

ChatGPT for Evaluation of Chart-editing

You are an expert evaluator tasked with assessing the performance of a model on a chart-editing task. You will be provided with the original code of the chart, the instructions given to the model, and the code generated by the model.

The original code: **[Original code]**
Instructions: **[Instructions]**
The generated code: **[Generated code]**

Using the criteria below, please score the model's performance:

Data Accuracy
0 points: The model makes no modifications to the chart based on the instructions.
1 point: The model makes some modifications to the chart, but they are largely incorrect based on the instructions.
2 points: The model makes modifications to the chart and some are correct, but key elements are missing or incorrect.
3 points: The model makes accurate modifications to the chart and most are correct based on the instructions.
4 points: The model makes accurate modifications to the chart and all are correct based on the instructions.
5 points: The model makes accurate and comprehensive modifications to the chart based on the instructions.

Completeness
0 points: The generated code is incomplete and shows no understanding of the original chart or instructions.
1 point: The generated code is partially complete but shows minimal understanding of the original chart and instructions.
2 points: The generated code is mostly complete but lacks some key elements or shows significant errors.
3 points: The generated code is largely complete with only minor elements missing or incorrect.
4 points: The generated code is almost entirely complete with only minor improvements possible.
5 points: The generated code is completely detailed and precise.

Aesthetics
0 points: The model fails to maintain or improve the aesthetics of the original chart.
1 point: The model makes some aesthetic modifications, but they are largely incorrect or inappropriate.
2 points: The model makes aesthetic modifications and some are correct, but key elements are missing or incorrect.
3 points: The model maintains or improves the aesthetics of the chart with some minor errors or omissions.
4 points: The model significantly enhances the aesthetics of the chart with only minor improvements possible.
5 points: The model excellently enhances the aesthetics of the chart with no improvements needed.

Instruction Following Performance
0 points: The model fails to follow the instructions at all.
1 point: The model follows some parts of the instructions but misses out on major aspects.
2 points: The model follows the instructions to a basic extent but misses out on or incorrectly interprets key elements.
3 points: The model largely follows the instructions with only minor elements missed or incorrectly interpreted.
4 points: The model follows the instructions almost entirely with only minor improvements possible.
5 points: The model follows the instructions excellently with no elements missed or incorrectly interpreted.

The output should first give the average score based on three criteria, then output scores for each criteria

Figure 14: **The prompt template used for evaluation on the Chart-editing task.** Input conditions include the original code corresponding to the given chart figure, the instructions that describe how to edit the figure, and the generated code. The output will contain the final average score and scores for each criterion.

ChatGPT for Evaluation of Chart-to-text

You are an expert evaluator assessing the performance of multi-modal LLM in generating detailed descriptions based on chart figure. Your task is to evaluate the description generated by multi-modal LLM. You receive the ground truth description and raw data for reference.
Here is the ground truth description for reference:
[Ground-truth description]

Here is the raw data:
[Raw data]

Now, here is the description generated by the multi-modal LLM:
[Predicted description]

Using the reference and generated descriptions above, please rate the performance of multi-modal LLM on a scale of 0 to 5 based on the following criteria:

0 Points:
- The generated description doesn't reference the chart data at all or is completely irrelevant.
- Multi-modal LLM doesn't show any understanding of the chart figures or raw data.

1 Point:
- The generated description refers to the chart, but most details are incorrect.
- Multi-modal LLM displays minimal understanding of the chart figures and raw data.

2 Points:
- The generated description refers to the chart, and some details are correct, but important elements are missing or incorrect.
- Multi-modal LLM shows basic understanding of the chart figures and raw data, but significant errors are present.

3 Points:
- The generated description accurately refers to the chart, and most details are correct.
- Multi-modal LLM shows a good understanding of the chart figures and raw data, but there are some minor errors or omissions.

4 Points:
- The generated description accurately refers to the chart, and all details are correct.
- Multi-modal LLM shows a strong understanding of the chart figures and raw data.
- There could be minor improvements in the clarity or completeness of the description.

5 Points:
- The generated description accurately and comprehensively describes the chart.
- Multi-modal LLM shows an excellent understanding of the chart figures and raw data, with no errors or omissions.
- The description is clear, detailed, and precise. It could be used as a standalone explanation of the chart.

Please provide your score in the first line and explain your rating in the second line.

Figure 15: **The prompt template used for evaluation on the Chart-to-text task.** The ground-truth description, raw data, and predicted description are input conditions. This evaluation prompt requires GPT-4 to give the final score and explanations of the given score.

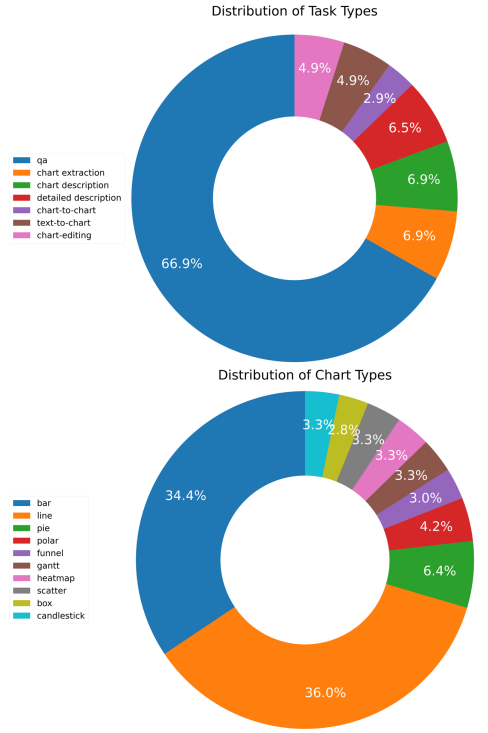


Figure 16: **Distributions of different types of data in our dataset**. The top and bottom pie charts show the distribution of task types and chart types, respectively. (The illustration is generated by our proposed ChartL-lama.)