Simplot: Enhancing Chart Question Answering by Distilling Essentials

Wonjoong Kim[*], **Sangwu Park**[*], **Yeonjun In**, **Seokwon Han**, **Chanyoung Park**[†]
KAIST
{wjkim, sangwu.park, yeonjun.in, dulwich10, cy.park}@kaist.ac.kr

Abstract

Recently, interpreting complex charts with logical reasoning has emerged as challenges due to the development of vision-language models. A prior state-of-the-art (SOTA) model has presented an end-to-end method that leverages the vision-language model to convert charts into table format utilizing Large Language Model (LLM) for reasoning. However, unlike natural images, charts contain a mix of essential and irrelevant information required for chart reasoning, and we discover that this characteristic can lower the performance of chart-to-table extraction. In this paper, we introduce Simplot, a method designed to extract only the elements necessary for chart reasoning. The proposed method involves two steps: 1) training to mimic a simple plot that contains only the essential information from a complex chart for table extraction, followed by 2) performing reasoning based on the table. Our model enables accurate chart reasoning without the need for additional annotations or datasets, and its effectiveness is demonstrated through various experiments[1].

1 Introduction

The rapid advancements in vision-language models have accelerated a wave of research into models capable of handling data that integrates both images and text (Zhang et al., 2021; Zhou et al., 2020; Kim et al., 2023, 2024), thus undertaking a variety of tasks. Among these tasks, the interest in models capable of advanced reasoning has significantly increased. This increasing field has seen considerable success in addressing visual question answering (VQA) tasks targeted at natural images, marking a trend towards models that can engage in complex reasoning based on images (Antol et al., 2015; Shao et al., 2023; Gardères et al., 2020). Despite these advancements, the domain of mathematical multimodal reasoning such as interpreting charts remains relatively unexplored. Mathematical reasoning in question answering poses unique challenges, as models proficient in natural images struggle with specific types, such as charts. Charts, with their unique formats and the need for logical interpretation, necessitate a different approach to learning compared to conventional VQA models targeting natural images.

Prior chart reasoning methods mainly rely on heuristic rule-based systems, and thus they are not only limited to pre-defined chart formats but also struggle with novel chart types without additional rule formulation (Luo et al., 2021). Moreover, the performance of models utilizing OCR or key-point detection modules are highly dependent on the performance of these modules, and also face significant annotation costs and are typically unable to perform end-to-end reasoning (Methani et al., 2020; Poco and Heer, 2017). In response to these limitations, recent approaches have adopted vision-language models trained in an end-to-end manner without heuristic rules (Cheng et al., 2023; Liu et al., 2022b), which, however, require fine-tuning for each specific downstream task, limiting their flexibility.

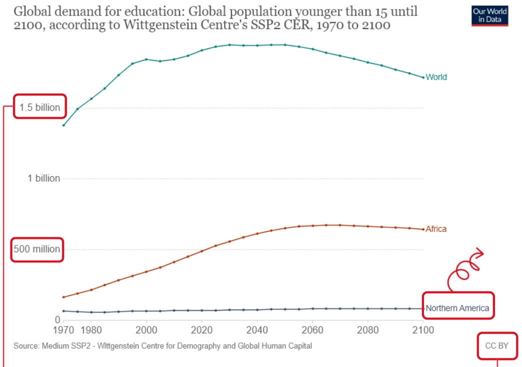
A novel approach, called Deplot (Liu et al., 2022a), that combines vision-language models with Large Language Models (LLMs) has emerged as an approach to addressing these issues. Specifically, Deplot first transforms charts into tables (i.e., chart-to-table extraction), and then employs the extracted tables along with the LLMs for reasoning. This approach not only aims to solve the inherent problems of previous methodologies, but also leverages the capability of the LLMs to enhance performance in chart question answering. Converting charts into tables before reasoning offers several advantages, including improved interpretability and the abil-

[*] These authors contributed equally.

[†] Corresponding author.

[1] Our source code is available at https://github.com/sangwu99/Simplot.

Findings of the Association for Computational Linguistics: NAACL 2025, pages 573–593
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics



(a) Input chart

Entity	1970	1980	...	2080	2100
World	*1.37*	*1.55*	...	*1.79*	*1.71*
Africa	190	200	...	650	637
CC BY	*1.3*	*1.3*	...	*1.3*	*1.3*

(b) Extracted table

Figure 1: An example of chart-to-table extraction (from (a) to (b)) by Deplot (Liu et al., 2022a) on ChartQA.

ity to achieve high performance in table reasoning, thus facilitates more accurate and precise reasoning compared to a direct image-based QA.

Despite its effectiveness, Deplot suffers from the following two limitations that still need to be addressed. First, we discovered that Deplot struggles to fully utilize the textual context within a chart for chart-to-table extraction (See Fig. 1). For example, the counting unit (e.g., “million” or “billion”) is overlooked when converting a chart (Fig. 1(a)) into a table (Fig. 1(b)). Moreover, irrelevant information (e.g., “CC BY”) is extracted as an entity in the table, while relevant information (e.g., “Northern America”) is overlooked. However, as real-world charts often contain information that might not be helpful for chart reasoning (e.g., source credit and “CC BY”), it is crucial for the model to be able to differentiate between relevant and irrelevant information. Owing to these challenges, Deplot is prone to eventually extracting inaccurate values within a table, failing in effectively interpreting the information of the chart.

The second limitation of Deplot is that it solely relies on tables to solve chart reasoning tasks, while overlooking the visual information associated with the tables. For this reason, Deplot fails to answer questions regarding the information that cannot be obtained from the extracted table itself (e.g., "What is the value of the third bar from the top?", "What year does the orange line represent?"), indicating a significant shortfall in the model’s capability to interpret visual data.

In this paper, we propose a simple yet effective method, named SIMPLOT, to address the aforementioned limitations of Deplot. To handle the first limitation, we develop two methods aimed at providing explicit supervision to the model. The first method, named row-col rendering, involves explicitly inserting information about the rows and columns that an extracted table should contain, allowing the model to perceive more accurate textual context. The second method involves conveying only the essential information in the chart to the model, which is built on our observation that converting an original chart containing irrelevant information into a simple chart containing only the essential information significantly improves the performance of table extraction.

To address the second limitation, we present a novel prompt named *Human-oriented chart instruction* while leveraging a Large Multimodal Model (LMM) to utilize the visual attributes of a chart. Since more advanced reasoning is required when interpreting charts, LMM requires a prompt that is specifically designed for the chart reasoning task, differentiated from natural images. Hence, we provide instructions to the model in a way that is similar to how humans interpret charts for precise reasoning.

Our contributions are summarized as follows:

1. We provide guidelines for utilizing textual information within charts for reasoning that was previously overlooked.
2. SIMPLOT extracts only essential information from complex charts, preventing irrelevant information from entering the model, resulting in detailed reasoning.
3. We present a prompt specifically designed for chart reasoning. Through this prompt, LMMs perform more accurate reasoning by mimicking how humans interpret charts.
4. Extensive experiments validate that SIMPLOT successfully addresses the limitations of Deplot. A further appeal of SIMPLOT is that it is model-agnostic, i.e., it can be applied to any existing model that involves chart-to-table extraction for chart reasoning beyond Deplot.

2 Proposed Method: SIMPLOT

In this section, we describe our proposed method, SIMPLOT. Fig 2 presents the overall framework

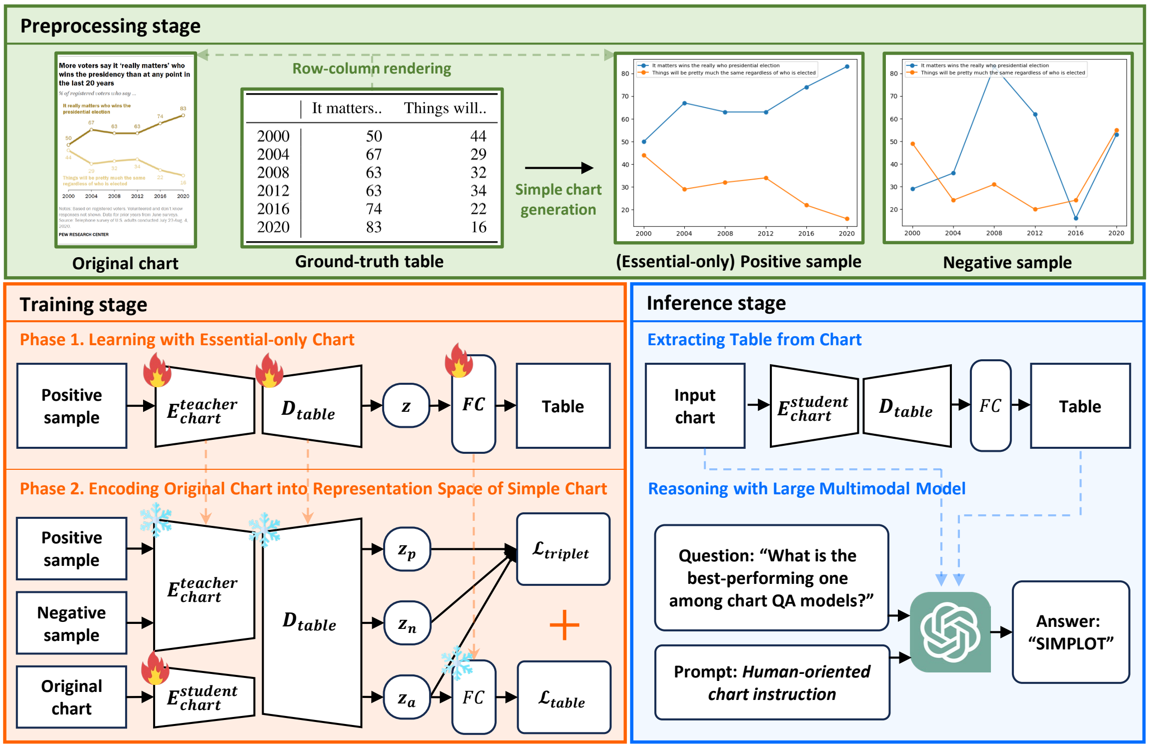

Preprocessing stage
More voters say it 'really matters' who wins the presidency than at any point in the last 20 years
Original chart
Row-column rendering
| It matters.. | Things will..
2000 50 44
2004 67 29
2008 63 32
2012 63 34
2016 74 22
2020 83 16
Ground-truth table
Simple chart generation
It matters wins the really who presidential election
Things will be pretty much the same regardless of who is elected
(Essential-only) Positive sample
Negative sample
Training stage
Phase 1. Learning with Essential-only Chart
Positive sample
$E^{teacher}_{chart}$
D_{table}
z
FC
Table
Phase 2. Encoding Original Chart into Representation Space of Simple Chart
Positive sample
Negative sample
Original chart
$E^{teacher}_{chart}$
$E^{student}_{chart}$
D_{table}
z_p
z_n
z_a
FC
$\mathcal{L}_{triplet}$
+
$\mathcal{L}_{table}$
Inference stage
Extracting Table from Chart
Input chart
$E^{student}_{chart}$
D_{table}
FC
Table
Reasoning with Large Multimodal Model
Question: "What is the best-performing one among chart QA models?"
Prompt: Human-oriented chart instruction
Answer: "SIMPLOT"

Figure 2: Overall framework of SIMPLOT. Upper box presents the preprocessing stage, which involves generating a simple positive sample containing only essential information from the original chart, as well as a negative sample, along with row and column rendering. Lower left box illustrates the training stage including two phases. In Phase 1 of the training stage, a teacher encoder and a table decoder are trained using a simple chart, and in Phase 2, a student encoder is trained with the original chart, while being distilled the knowledge of the teacher encoder on how to generate a table from a simple chart. Lower right box illustrates the inference stage, where an LMM receives the original chart and the extracted table along with prompts for reasoning.

of SIMPLOT, where SIMPLOT eventually performs reasoning with an LMM given a table extracted from a chart (Inference stage). In this work, we focus on chart-to-table extraction (Training stage) as a high-quality table enables precise reasoning with a powerful LMM. It is important to note that SIMPLOT can be combined with diverse LMM variants, and various techniques such as Chain-of-Thought (CoT) (Wei et al., 2022) can further improve the accuracy of the chart reasoning, which we leave as future work.

2.1 Preprocessing Stage

Before the training stage, we conduct two preprocessing steps, i.e., 1) Simple Chart Generation, and 2) Row-Column Rendering. Note that as these steps can be readily done offline, they do not increase the training time.

1) Simple Chart Generation. We generate a simple chart by excluding irrelevant information from the original chart, keeping only essential elements required for reasoning. Specifically, since each of the original charts in the dataset is annotated with a table in the CSV format, we use a Python library (i.e., Matplotlib) to plot a chart based on the table. The chart generated in this manner is considered as a simplified version of the original chart, i.e., simple chart. This process requires no separate training or additional costs and it can be executed offline by running a simple code snippet, so it is scalable and generalizable across various datasets.

2) Row-Column Rendering. Inspired by rendering questions over images used for QA (Lee et al., 2023), we aim to improve the table extraction accuracy by rendering information about the rows and columns that should be included in the table onto the image, enabling the model to utilize this information to extract more accurate tables when converting charts to tables. Note that in the training stage, since we are given the ground-truth chart-table pairs, we simply render rows and columns of each table onto its paired image containing the chart. However, it would not be feasible in the inference stage, since we would be only given the charts without tables. Instead, in the inference stage, we utilize a LMM to extract rows and columns from the chart and render them onto the image. Despite the relative low performance of LMM such as GPT-4 (Achiam et al., 2023) in chart reasoning, we found they can accurately extract row and column information from charts, since such information is given as text and structured in a relatively simple manner. Detailed description of the process is presented in Appendix B.

2.2 Training Stage: Chart-to-Table Extraction

Our proposed chart-to-table extraction approach consists of two phases: **Phase 1)** Training a teacher encoder and a table decoder by performing the chart-to-table extraction given simple charts containing only the essential information for reasoning, rather than complex original charts; **Phase 2)** Training a student encoder by extracting the table given the original chart, while being distilled the knowledge by teacher encoder to embed original charts to the embedding space of simple charts.

2.2.1 Phase 1: Learning with Essential Part from Simple Chart

Our model learns the process of extracting tables from previously generated simple charts. Specifically, we use Deplot (Liu et al., 2022a) as our backbone model, which is our baseline model that consists of an image encoder and a text decoder, and fine-tune it on the generated simple chart-table pairs. This facilitates the image encoder to obtain representations containing only essential information within the chart, while the text decoder converts this representation into a table format. In this paper, we name the encoder and decoder as *chart encoder* E_{chart} and *table decoder* D_{table}, respectively.

To corroborate our model design of focusing on the essential parts when learning from charts, in Fig. 3, we compare the chart-to-table extraction performance when using either one of the two chart types (i.e., original chart (in green), and simple chart containing essential information only (in red)) for both training and inference. We observe that using simple charts for chart-to-table extraction greatly outperforms the case when original charts are used.

Note that the chart encoder trained here serves as the teacher encoder $E_{chart}^{teacher}$, providing guidelines for the representation that the student encoder $E_{chart}^{student}$ should learn in the subsequent learning process to be described in Sec. 2.2.2. Following Deplot, we adopt ViT (Dosovitskiy et al., 2020) as the chart encoders. Note that the trained table decoder D_{table} and a fully connected layer (FC) remain frozen in the next stage.

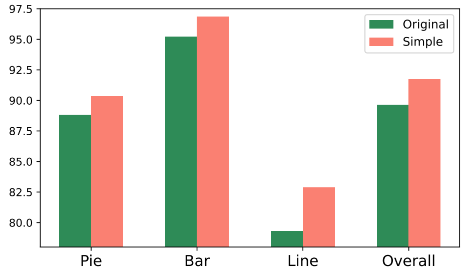


Figure 3: Accuracy of chart-to-table extraction using the original (in green) and simple charts (in red).

2.2.2 Phase 2: Encoding Original Chart into Representation Space of Simple Chart

As shown in Fig. 2, after training the teacher encoder $E_{chart}^{teacher}$ with simple charts as inputs, our goal is to train a student encoder $E_{chart}^{student}$ in a way that the representation of an original chart image obtained from $E_{chart}^{student}$ closely matches the representation of its corresponding simplified chart obtained from $E_{chart}^{teacher}$. This is to enable $E_{chart}^{student}$ to encode any original chart into the representation space of simplified chart images, thereby improving the accuracy of table generation. Then, given the chart representations, we generate table representations z using the frozen table decoder D_{table} trained in Phase 1.

More precisely, given an original chart A and the simplified chart P, we obtain the representations of the original chart z_a and its corresponding sample z_p from sequentially passing into chart encoders and the table decoder. For efficient learning, we define the triplet loss by adding a negative sample N, which is created by randomly shuffling the values of the simple chart P and its representation z_n as follows:

$$\mathcal{L}_{triplet}(A, P, N) = max\{d(z_a, z_p) - d(z_a, z_n) + m, 0\},$$

$$\text{where} \quad \begin{aligned} z_a &= D_{table}(E_{chart}^{student}(A)), \\ z_p &= D_{table}(E_{chart}^{teacher}(P)), \\ z_n &= D_{table}(E_{chart}^{teacher}(N)). \end{aligned} \qquad (1)$$

where d denotes the distance defined as $d(z_i, z_j) = \|z_i - z_j\|_2$, and m denotes the margin which induces the anchor z_a to be closer to the positive sample z_p and farther from the negative sample z_n. The purpose of utilizing negative samples is to more precisely discriminate between data that may be mapped closely in the representation space because the charts are similar, and further details and effectiveness are presented in Appendix H.

Through this process, the student chart encoder $E_{chart}^{student}$ is trained to extract only the essential information from original charts, even if they contain information that is irrelevant to chart reasoning. The representation z_a obtained after passing the original chart A through the student encoder $E_{chart}^{student}$ and the table decoder D_{table} is then fed into the frozen fully connected layer (FC) trained in Phase 1 to generate a table.

Furthermore, since our goal is not just to train $E_{chart}^{student}$ to mimic the representation of simple charts, but to eventually generate tables effectively, we also introduce a cross-entropy loss for table generation as follows:

$$T = [\hat{y}_1, \ldots, \hat{y}_n] = FC(z_a), \tag{2}$$

$$\mathcal{L}_{table} = -\frac{1}{N} \sum_{i=1}^{n} \sum_{c=1}^{C} y_{i,c} \log \left(\frac{\exp(\hat{y}_{i,c})}{\sum_{j=1}^{C} \exp(\hat{y}_{i,j})} \right), \tag{3}$$

where T a linearized textual sequence of the generated table, n and C denote the length of the linearized textual sequence of the table T, and the number of classes, respectively, and y_i and $\hat{y}_i$ denote the i-th ground-truth and the predicted token each belonging to one of the C classes, i.e., $y_{i,c} = 0$ if y_i belongs to class c and 0 otherwise. Note that, as in Deplot, lines of T are separated by '<0x0A>' (line break), and cells are distinguished by '|'.

The final loss function for chart-to-table extraction is defined as follows:

$$\mathcal{L}_{final} = \lambda \mathcal{L}_{triplet} + (1 - \lambda) \mathcal{L}_{table}. \tag{4}$$

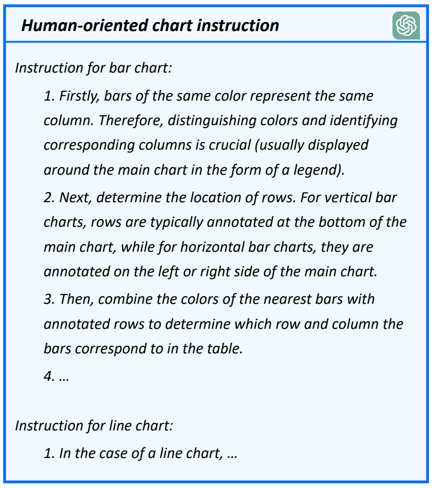
where λ is a hyperparameter balancing the two losses, i.e., $\mathcal{L}_{triplet}$ and $\mathcal{L}_{table}$. Note that considering the scale of the two losses, we set λ to 0.1 to balance between them. For detailed analysis of hyperparameter λ, please refer to Appendix G.

In summary, in Phase 2, we align the representation space of $E_{chart}^{student}$, which is trained with original charts, and $E_{chart}^{teacher}$, which is trained with simple charts, thereby allowing $E_{chart}^{student}$ to generate chart representations that mainly contain essential information for chart reasoning. Based on the representations obtained from $E_{chart}^{student}$, we use the frozen decoder D_{table} to generate tables that will be passed on to the LMM in the inference stage.

2.3 Inference Stage: Reasoning with Extracted Table

In the inference stage, we use a Large Multimodal Model (LMM) to perform various reasoning tasks given the tables obtained from the training stage as input. However, relying solely on tables without the associated visual information to solve chart reasoning tasks makes it impossible for the model to answer questions regarding visual information in charts (e.g., "What is the value of the third bar from the top?", "What year does the orange line represent?"). To address this issue, we additionally provide the original chart as another input to the LMM to enable the model to provide answers to a wider range of questions. By doing so, we expect the model to not only answer to questions regarding precise chart values by referring to the tables, but also to answer about visual attributes not contained in the table by referring to the original chart. Please refer to Appendix F for experimental results on the effects of using the table and original chart together.

Human-oriented chart instruction. Furthermore, we present a novel prompt that is specifically designed for chart reasoning, named *Human-oriented chart instruction*. Although prompt engineering is a widely studied topic for processing natural images (Kim et al., 2023; Li et al., 2022), it is crucial to develop specialized prompts specifically tailored for charts, due to the inherent nature of chart reasoning tasks, i.e., they require more advanced advanced thinking steps for interpretation. Our proposed prompt provides instructions that mimic how humans interpret charts, enabling more accurate reasoning within the LMM and its effec-



Human-oriented chart instruction

Instruction for bar chart:

1. Firstly, bars of the same color represent the same column. Therefore, distinguishing colors and identifying corresponding columns is crucial (usually displayed around the main chart in the form of a legend).

2. Next, determine the location of rows. For vertical bar charts, rows are typically annotated at the bottom of the main chart, while for horizontal bar charts, they are annotated on the left or right side of the main chart.

3. Then, combine the colors of the nearest bars with annotated rows to determine which row and column the bars correspond to in the table.

4. ...

Instruction for line chart:

1. In the case of a line chart, ...

Figure 4: A snippet of the proposed prompt.

tiveness will be demonstrated in Section 3.1.2. We provide a snippet of our proposed prompt in Fig. 4. For detailed description of *Human-oriented chart instruction*, please refer to Appendix C.

3 Experiments

Dataset. In this paper, we evaluate SIMPLOT using widely used dataset for chart reasoning, ChartQA (Masry et al., 2022) comprising three types of charts (i.e., pie, bar, and line), and PlotQA (Methani et al., 2020) including three types of charts (i.e., dot line, line, bar). ChartQA also includes human-authored and LLM-augmented QA pairs, while PlotQA consists of QA pairs generated from templates created by manually analyzed sample questions collected through crowd-sourcing. Detailed explanation for dataset is presented in Appendix D.1.

Baselines. In this paper, we use various models as baselines that can perform chart reasoning. The baselines we employ are divided into three categories including Vision-Language pre-trained models (VLP), fully supervised models for chart reasoning, and models which extract table and perform reasoning using LLMs. For a fair comparison, we employ GPT-4V for all methods utilizing extracted table. Detailed descriptions of the models are provided in the Appendix D.2.

Evaluation protocol. For question answering based on charts, we report Relaxed Accuracy (RA) of 2,500 questions in the test set following previous works (Lee et al., 2023; Kantharaj et al., 2022b; Masry et al., 2022; Kantharaj et al., 2022a; Liu et al., 2022b).

Furthermore, we propose a proper metric to evaluate the chart-to-table extraction performance, employing a slightly modified metric derived from Relative Mapping Similarity (RMS), presented by Deplot (Liu et al., 2022a). We discover that the vanilla RMS does not accurately measure the chart-to-table performance in certain cases, and thus we introduce a metric named Relative Distance (RD). For the differences between RMS and RD, please refer to the Appendix D.3.

3.1 Experimental Results

3.1.1 Results on Chart-to-Table Extraction

In Table 1, we observe that SIMPLOT exhibits superior chart-to-table extraction performance compared to existing methods including Deplot (Liu et al., 2022a) and UniChart (Masry et al., 2023).

Models	Chart type			Overall
	Pie	Bar	Line	
GPT-4V	90.13	91.53	71.51	84.24
UniChart	84.86	92.58	**85.16**	88.03
Deplot	88.82	96.37	82.25	90.95
SIMPLOT	**91.41**	**96.87**	84.74	**92.32**

Table 1: Chart-to-table extraction performance (RD_{F1}) on the ChartQA dataset over various chart types.

We argue that the superiority of SIMPLOT in terms of chart-to-table extraction eventually leads to more precise reasoning as will be shown in Section 3.1.2. While the difference in chart-to-table performance may seem small, we argue that even minor errors such as slightly incorrect values or column and row names during the chart-to-table extraction may entail completely incorrect answers, as shown in Table 2.

We also have mentioned that GPT-4V can achieve more accurate text extraction rather than numerical extraction, which is why we utilize it for row and column extraction. Nevertheless, to further validate the necessity of a model specifically targeting charts, we also conduct table extraction (both text and numerical information) using GPT-4V, and the results are presented in the Table 1. As previously argued, GPT-4V is well-suited for extracting text information from charts, so it demonstrated comparable performance on pie charts where values and rows/columns are primarily given as text. However, for line charts, where most values must be inferred from the image rather than being provided as text, GPT-4V showed significantly lower performance compared to other models.

This explains the low QA performance in Table 2 as well, and conducting QA using tables extracted by GPT-4V is inefficient both in terms of performance and cost. Furthermore, these results support our argument that existing vision-language models still lack effectiveness when dealing with chart images while performing well on natural images, so methods specifically trained to target charts are necessary.

3.1.2 Results on Chart Reasoning

The performance of question answering on the ChartQA dataset is summarized in Table 2. 1) In general, we observe that methods designed with a focus on chart reasoning (i.e., 'Supervised' and

[2] The authors also provide pre-trained model for chart to table extraction. We use this model to extract the table and conduct reasoning in the same setting of Deplot and SIMPLOT.

	Models	Data type		
		Human	Augmented	Overall
VLP models	TaPas	28.72	53.84	41.28
	V-TaPas	29.60	61.44	45.52
	T5	25.12	56.96	41.04
	VL-T5	26.24	56.88	41.56
	PaLI	30.40	64.90	47.65
	Mini-GPT	8.40	15.60	12.00
	LLaVa	37.68	72.96	55.32
	GPT-4V	56.48	63.04	59.76
Supervised	ChartQA	40.08	63.60	51.84
	ChartT5	31.80	74.40	53.10
	Pix2Struct	30.50	81.60	56.05
	MatCha	38.20	90.20	64.20
	Unichart	43.92	88.56	66.24
	ChartLlama	48.96	90.36	69.66
	ChartAssisstant	65.90	**93.90**	79.90
	ChartInstruct	45.52	87.76	66.64
Table	Deplot	62.71	78.63	70.67
	Unichart[2]	67.04	69.92	68.48
	SIMPLOT	**78.07**	88.42	**83.24**

Table 2: Chart question answering performance (RA) on the ChartQA dataset.

'Table') outperform VLP in QA tasks. This demonstrates that despite the superior performance of vision-language models in performing a wide range of tasks in the traditional natural image domain, they face difficulties in interpreting charts due to the significantly different characteristics of natural images and charts. 2) Furthermore, noticeable performance difference is observed among models aimed at chart reasoning. Methods using tables such as Deplot, Unichart, and SIMPLOT, generally outperform models that conduct reasoning based solely on images (i.e., 'Supervised'). This suggests that going through tables allows for more detailed reasoning, aligning well with our motivation that emphasizes the importance of effectively extracting tables. Additionally, models utilizing extracted tables and LMM can efficiently handle various tasks such as OpenCQA (Kantharaj et al., 2022a) without any additional data and training. 3) Among the table-based reasoning models, SIMPLOT performs the best by utilizing textual information, which is essential for obtaining the most accurate table, and by employing prompts mimicking the human reasoning process. Additional experimental results on PlotQA dataset and effectiveness of our method are presented in Appendix E and 3.2, respectively.

Discussion. The overall performance of SIMPLOT is consistently higher than other baselines; however, on the augmented set, it sometimes exhibits slightly lower performance compared to some models. This can be attributed to the characteristic of the augmented set, which is obtained by fine-tuning the T5 model on the SQuAD QA dataset (Rajpurkar et al., 2016) to generate questions and answers. Consequently, the questions in the augmented set are mostly simpler and relatively easier to answer compared to those in the human set. Models that are fully fine-tuned on such questions tend to show notably higher performance on augmented data. For instance, MatCha (Liu et al., 2022b) shows a performance of 90.2 on the augmented set, but its performance drops significantly to 38.2 on the human set which is composed of more complex QA pairs, resulting in an overall performance of 64.2. In contrast, SIMPLOT's high performance even on the human set without additional fine-tuning suggests that proposed method is effective for more complex reasoning tasks.

3.2 Further Analysis

Ablation Study. We conduct ablation studies to determine the impact of each module of SIMPLOT on the performance. Upper part of the Table 3 shows the effects of the two modules we introduced for chart-to-table extraction: 1) row-column rendering and 2) distillation from a simple chart. These results demonstrate that each module enhances the performance of table extraction, thereby proving the effectiveness of our design.

Additionally, to verify the effect of the proposed prompt, we also present the QA performance with and without the *Human-oriented chart instruction* (lower part of Table 3). We observe that our prompt that mimics how humans think enhances the performance of the QA task, which confirms the importance of using appropriate prompts tailored to the task when employing LMMs. It is notable that providing more diverse in-context examples could lead to even greater performance gains.

Row-col rendering	Simple chart	Prompt	RD_{F1}	RA
✗	✗	-	90.95	-
✓	✗	-	91.40	-
✗	✓	-	91.86	-
✓	✓	-	**92.32**	-
-	-	✗	-	79.79
-	-	✓	-	**83.24**

Table 3: Ablation studies on each component of SIMPLOT for chart-to-table extraction and QA tasks.

SIMPLOT is Model-agnostic. While SIMPLOT is originally designed to address the limitations of Deplot, it is model-agnostic, i.e., SIMPLOT can be applied to any existing table-based reasoning model to enhance its performance. Here, we applied SIMPLOT to one of our baselines, Unichart

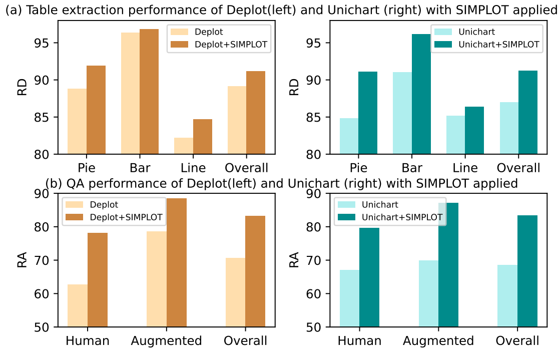


Figure 5: Improvements in performance observed when SIMPLOT is applied across diverse models.

(Masry et al., 2023), which is pre-trained to perform chart-to-table extraction task. Specifically, Fig. 5 (a) demonstrates significant performance enhancements with SIMPLOT applied to Deplot and Unichart, respectively. Consequently, Fig. 5 (b) show a substantial increase in question answering performance. These results verifies the generality and practicality of SIMPLOT.

Using Images as Input to Unichart/Deplot for Inference. Here, we further demonstrate that using both tables and images is essential for chart reasoning, as emphasized earlier. Table 4 shows that the performance of the vanilla Deplot and the vanilla Unichart, both of which only use tables for reasoning, is improved when images are also utilized during inference. This indicates a clear drawback of relying solely on tables and proves the necessity of using images with specifically designed prompts. Moreover, even without the prompt, SIMPLOT outperforms the baselines with images used during inference, thanks to the superior table extraction capability of SIMPLOT.

Models	Human	Augmented	Overall
Unichart	67.04	69.92	68.48
Unichart + img.	75.04	**88.82**	81.93
Unichart + SIMPLOT w/o prompt	76.56	88.64	82.60
Unichart + SIMPLOT	**79.56**	87.18	**83.37**
Deplot	62.71	78.63	70.67
Deplot + img.	72.39	85.01	78.70
Deplot + SIMPLOT w/o prompt	73.91	85.67	79.79
Deplot + SIMPLOT	**78.07**	**88.42**	**83.24**

Table 4: Chart question answering performance (RA) with/without image on ChartQA dataset.

Broad Applicability of the Proposed Prompt. To verify the effectiveness of our proposed prompt, we apply it to Deplot with images (upper part of Table 5). We observe that applying the proposed prompt to the 'Deplot+img' indeed greatly enhances its performance.

Performance on more Challenging Settings. It is worth noting that 'Deplot+img+prompt' now performs competitively to SIMPLOT. However, we argue that this is mainly due to the simplicity of the ChartQA dataset, which makes the dataset insufficient to evaluate the table extraction capability of SIMPLOT. For example, ChartQA contains questions that can be answered by referencing just one row of the extracted table.

To evaluate the models with more challenging questions, we randomly sample 100 test images and then provide GPT-4 with the following instruction: *"Create a challenging question-answer pair that requires referencing at least two rows and two columns to solve."*. Through this process, we generated 100 question-answer pairs and eventually obtained 85 pairs after manually filtering out inaccurate pairs, and then evaluate 'Deplot+img+prompt' and SIMPLOT on these question-answer pairs (lower part of Table 5). We observe that the performance gap greatly increases as the questions get more challenging, which demonstrates the importance of extracting accurate tables when encountering challenging questions. For the specific example of the challenging question and how the extracted tables from SIMPLOT and Deplot lead to differences in complex QA performance, please refer to Appendix J.

	Models	Human	Augmented	Overall
Easy	Deplot + img.	72.39	85.01	78.70
	Deplot + img. + prompt	77.75	88.30	83.03
	SIMPLOT	**78.07**	**88.42**	**83.24**
Hard	Deplot + img. + prompt	-	-	49.41
	SIMPLOT	-	-	**65.88**

Table 5: Performance on more challenging questions.

Models	Com. (Easy)	Com. (Hard)	Sequential
Deplot+img.+prompt	57.50	50.00	26.80
SIMPLOT	**59.20**	**58.10**	**36.50**

Table 6: Performance on MultiChartQA dataset.

Furthermore, questions that necessitate referring to multiple charts (Zhu et al., 2024) are expected to be significantly impacted by the accuracy of table extraction when it comes to improving QA performance. The additional experimental results are presented in the Table 6. On the comparative dataset, which requires simultaneously comparing values from two or more charts, and the sequential dataset, which requires finding new information in one chart based on information from another chart, our proposed method consistently outperforms Deplot. Since these tasks require examining multiple charts simultaneously, the results strongly support our claim that accurate table extraction leads to better performance on complex tasks. Moreover, the comparative dataset is divided into easy and hard categories, and the greater performance differences observed on harder questions indicate that even small improvements in table extraction accuracy can significantly impact accurate QA.

Adaptability of Chart-to-table Extraction to another task. As we have argued before, extracting tables and combining them with LMM not only improves the accuracy of QA but also offers the advantage of immediately being applicable to other tasks such as OpenCQA (Kantharaj et al., 2022a) without additional training.

Table 7 presents the BLEU score (Papineni et al., 2002) of a model fine-tuned for OpenCQA (ChartInstruct), and those of models that utilize extracted tables (Deplot & SIMPLOT). Here, we demonstrate that leveraging tables allows for effective performance on other tasks without additional training, while also proving that accurately extracting tables enhances accuracy when performing these tasks. Note that this dataset includes charts in a different format from those used to train SIMPLOT, so training with the charts from this dataset could potentially improve performance.

Models	BLEU score
Gemini	6.84
GPT-4V	3.31
ChartInstruct	5.27
Deplot	6.88
SIMPLOT	**7.11**

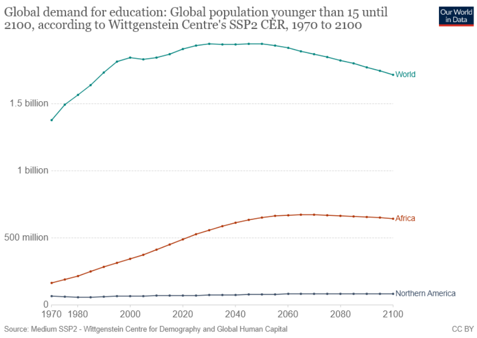
Table 7: Open-ended question answering performance on OpenCQA dataset.

Case Studies. We verify whether SIMPLOT correctly captures textual information within charts. Fig. 6 illustrates that the table extracted by Deplot fails to consider the counting unit, i.e., "million/billion," present in the chart, while SIMPLOT correctly captures the textual information.

Please refer to Appendix I for further case studies including the results that demonstrate the effectiveness of proposed prompt and Appendix K for error analysis.

4 Related Work

Early chart processing methods relied on rule-based approaches, which lacked flexibility for various chart types (Balaji et al., 2018; Choi et al., 2019). OCR and key-point detection methods are utilized to address these limitations, but they depend on data annotation and module performance, making them time-consuming (Poco and Heer, 2017; Zhou et al., 2023; Xue et al., 2023).



(a) Input chart

Entity	1970	1980	2000	2020	2040	2060	2080	2100
World	1.37	1.55	1.83	1.93	1.93	1.89	1.79	1.71
Africa	190	200	350	479	604	665	650	637
Northern America	6	3.79	3.23	3.92	4.06	4.63	4.79	4.81

Entity	1970	1980	2000	2020	2040	2060	2080	2100
World	1388.75	1520.46	1830.13	1934.69	1952.49	1970.42	1712.03	1696.22
Africa	117.45	115.97	334.05	490.55	642.36	677.90	686.98	694.67
Northern America	0.06	0.05	0.059	0.05	0.05	0.06	0.06	0.06

(b) Extracted table from Deplot (upper) & SIMPLOT (lower)

Figure 6: SIMPLOT captures contextual information (*billion/million*) and extract precise values in the chart.

Recent papers train end-to-end vision-language models to improve chart reasoning without heuristic rules, but they need fine-tuning for specific tasks (Meng et al., 2024; Huang et al., 2023; Masry et al., 2023; Han et al., 2023), so recent approaches transform charts into tables and utilized LLM for question answering while enhancing interpretability and allowing the model apply to various downstream task without additional training (Liu et al., 2022a; Xia et al., 2023). For complete related works, please refer to the Appendix A.

5 Conclusion

We propose SIMPLOT, which extracts only the essential information required for chart reasoning. Furthermore, by leveraging textual information inside the charts, SIMPLOT enables accurate chart reasoning. We also propose a novel prompt specifically designed for chart reasoning, which guides LMM to imitate how humans interpret chart. Through extensive experiments, we demonstrate that SIMPLOT effectively handle concerns raised by existing works while improving performance, and also can be applied to other various tasks.

Limitations

As with existing chart reasoning methods, SIMPLOT is also expected to have limited performance on unseen out-of-distribution (OOD) data. This is a future research area that all chart reasoning methods, including ours, should consider. However, please note that our proposed method can address various downstream tasks with just one round of training to extract tables from OOD charts, unlike other methods that require separate training for each task. Meanwhile, we conduct inference without training for the OpenCQA dataset where no tables were provided, and we observed that SIMPLOT outperformed the existing baselines, indicating that our proposed method ensures a certain level of generalization performance for OOD charts.

Ethics Statement

SIMPLOT has not been trained with private or sensitive data. This significantly reduces the risk of generating harmful or misleading content. ChartQA and PlotQA dataset, which we utilize to collect chart-table pairs, is publicly accessible for research purposes. To make sure the transparency and reproducibility of our experiments, we provide detailed information on hyperparameter configurations in our paper and publicly share our source code. This careful approach mitigates the potential for unethical outcomes associated with data usage. While our models demonstrate state-of-the-art performance on the both dataset, we acknowledge the possibility of their misuse. There exists a risk that our models could be exploited to mislead the public about the content and implications of charts. Despite our models' high performance, we cannot guarantee their outputs will always be accurate, emphasizing the need for critical interpretation and verification of results.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2023-00216011, Development of artificial complex intelligence for conceptually understanding and inferring like human) and National Research Foundation of Korea(NRF) funded by Ministry of Science and ICT (NRF-2022M3J6A1063021).

References

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

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

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Abhijit Balaji, Thuvaarakkesh Ramanathan, and Venkateshwarlu Sonathi. 2018. Chart-text: A fully automated chart image descriptor. *arXiv preprint arXiv:1812.10636*.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Minesh Mathew, CV Jawahar, Ernest Valveny, and Dimosthenis Karatzas. 2019. Icdar 2019 competition on scene text visual question answering. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1563–1570. IEEE.

Hermann Bujard, Reiner Gentz, Michael Lanzer, Dietrich Stueber, Michael Mueller, Ibrahim Ibrahimi, Marie-Therese Haeuptle, and Bernhard Dobberstein. 1987. [26] a t5 promoter-based transcription-translation system for the analysis of proteins in vitro and in vivo. In *Methods in enzymology*, volume 155, pages 416–433. Elsevier.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022a. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. 2022b. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer.

Zhi-Qi Cheng, Qi Dai, Siyao Li, Jingdong Sun, Teruko Mitamura, and Alexander G Hauptmann. 2023. Chartreader: A unified framework for chart derendering and comprehension without heuristic rules. *arXiv preprint arXiv:2304.02173*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR.

Jinho Choi, Sanghun Jung, Deok Gun Park, Jaegul Choo, and Niklas Elmqvist. 2019. Visualizing for the non-visual: Enabling the visually impaired to use visualization. In *Computer Graphics Forum*, volume 38, pages 249–260. Wiley Online Library.

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

François Gardères, Maryam Ziaeefard, Baptiste Abeloos, and Freddy Lecue. 2020. Conceptbert: Concept-aware representation for visual question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 489–498.

Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. 2023. Chartllama: A multimodal llm for chart understanding and generation. *arXiv preprint arXiv:2311.16483*.

Johan Holmgren, Paul Davidsson, Jan A Persson, and Linda Ramstedt. 2012. Tapas: A multi-agent-based model for simulation of transport chains. *Simulation Modelling Practice and Theory*, 23:1–18.

Kung-Hsiang Huang, Mingyang Zhou, Hou Pong Chan, Yi R Fung, Zhenhailong Wang, Lingyu Zhang, Shih-Fu Chang, and Heng Ji. 2023. Do lvlms understand charts? analyzing and correcting factual errors in chart captioning. *arXiv preprint arXiv:2312.10160*.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.

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

Shankar Kantharaj, Xuan Long Do, Rixie Tiffany Ko Leong, Jia Qing Tan, Enamul Hoque, and Shafiq Joty. 2022a. Opencqa: Open-ended question answering with charts. *arXiv preprint arXiv:2210.06628*.

Shankar Kantharaj, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. 2022b. Chart-to-text: A large-scale benchmark for chart summarization. *arXiv preprint arXiv:2203.06486*.

Kibum Kim, Kanghoon Yoon, Yeonjun In, Jinyoung Moon, Donghyun Kim, and Chanyoung Park. 2024. Adaptive self-training framework for fine-grained scene graph generation. *arXiv preprint arXiv:2401.09786*.

Kibum Kim, Kanghoon Yoon, Jaehyeong Jeon, Yeonjun In, Jinyoung Moon, Donghyun Kim, and Chanyoung Park. 2023. Llm4sgg: Large language model for weakly supervised scene graph generation. *arXiv e-prints*, pages arXiv–2310.

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.

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

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. 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*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023a.

Mathvista: Evaluating math reasoning in visual contexts with gpt-4v, bard, and other large multimodal models. *arXiv e-prints*, pages arXiv–2310.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023b. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv preprint arXiv:2304.09842*.

Junyu Luo, Zekun Li, Jinpeng Wang, and Chin-Yew Lin. 2021. Chartocr: Data extraction from charts images via a deep hybrid framework. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1917–1925.

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 Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

Ahmed Masry, Mehrad Shahmohammadi, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. 2024. Chartinstruct: Instruction tuning for chart comprehension and reasoning. *arXiv preprint arXiv:2403.09028*.

Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2024. Chartassisstant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. *arXiv preprint arXiv:2401.02384*.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Jorge Poco and Jeffrey Heer. 2017. Reverse-engineering visualizations: Recovering visual encodings from chart images. In *Computer graphics forum*, volume 36, pages 353–363. Wiley Online Library.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Zhenwei Shao, Zhou Yu, Meng Wang, and Jun Yu. 2023. Prompting large language models with answer heuristics for knowledge-based visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14974–14983.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

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

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*.

Wenyuan Xue, Dapeng Chen, Baosheng Yu, Yifei Chen, Sai Zhou, and Wei Peng. 2023. Chartdetr: A multi-shape detection network for visual chart recognition. *arXiv preprint arXiv:2308.07743*.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5579–5588.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason Corso, and Jianfeng Gao. 2020. Unified vision-language pre-training for image captioning and vqa. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13041–13049.

Mingyang Zhou, Yi R Fung, Long Chen, Christopher Thomas, Heng Ji, and Shih-Fu Chang. 2023. Enhanced chart understanding in vision and language task via cross-modal pre-training on plot table pairs. *arXiv preprint arXiv:2305.18641*.

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*.

Zifeng Zhu, Mengzhao Jia, Zhihan Zhang, Lang Li, and Meng Jiang. 2024. Multichartqa: Benchmarking vision-language models on multi-chart problems. *arXiv preprint arXiv:2410.14179*.

A Complete Related Work

Pre-trained vision-language models. Vision-language pre-training seeks to enhance the performance of downstream vision and language tasks such as visual question answering (VQA) (Tan and Bansal, 2019; Alayrac et al., 2022), image-text retrieval (Chen et al., 2020; Jia et al., 2021), and image captioning (Li et al., 2020, 2022).

Several models learn cross-modal representation from visual features and language tokens (Chen et al., 2020; Su et al., 2019). Specifically, Lu et al. (2019) employs two single-modal networks, alongside a cross-modal transformer layer integrating information from input sentences and images.

Others propose a different approach to develop an enhanced Large Multimodal Model (LMM). For example, Liu et al. (2023) employs a visual encoder and a language decoder from other studies (Radford et al., 2021; Chiang et al., 2023). It is capable of performing various QA tasks including conversation, detail description, complex reasoning and focuses on chat capability.

Chart reasoning models. Early methods for processing charts predominantly utilized rule-based approaches (Balaji et al., 2018; Choi et al., 2019). However, they were specifically tailored to predefined chart formats, limiting their applicability only to certain types of charts. Moreover, incorporating new chart designs require additional rules, leading to the impracticality of immediate adaptation to new chart formats.

Due to above limitations, researchers utilize modules such as OCR and key-point detection (Poco and Heer, 2017; Zhou et al., 2023; Xue et al., 2023). However, these methods can be time-consuming and require dataset annotation for labeling, and rely on the performance of modules. For example, Luo et al. (2021) suffers from a drawback as it solely predicts the raw data values without establishing connections to their respective axes or legends.

In an effort to address these challenges, researchers develop vision-language models (Meng et al., 2024; Huang et al., 2023) without heuristic rules. Some approaches comprehend the chart and respond to questions in natural language (Masry et al., 2023; Han et al., 2023). However, such models require fine-tuning for each downstream task, constraining their adaptability for diverse tasks. Meanwhile, studies such as Methani et al. (2020); Lu et al. (2023a) have created datasets to enable more realistic mathematical reasoning.

To tackle these challenges, recent studies Liu et al. (2022a); Xia et al. (2023) introduce a two-step approach, chart-to-table extraction and reasoning with LLM. Recent advancements like Liu et al. (2022a); Xia et al. (2023) have introduced a two-step approach to tackle challenges in chart analysis. Firstly, charts are transformed into tables, improving interpretability and aiding in identifying inaccuracies. Lu et al. (2023b) utilizes Program of Thought (PoT), achieving significant performance gains. This approach enables more accurate and interpretable reasoning compared to traditional methods, indicating that table extraction is crucial.

B Row-column Rendering

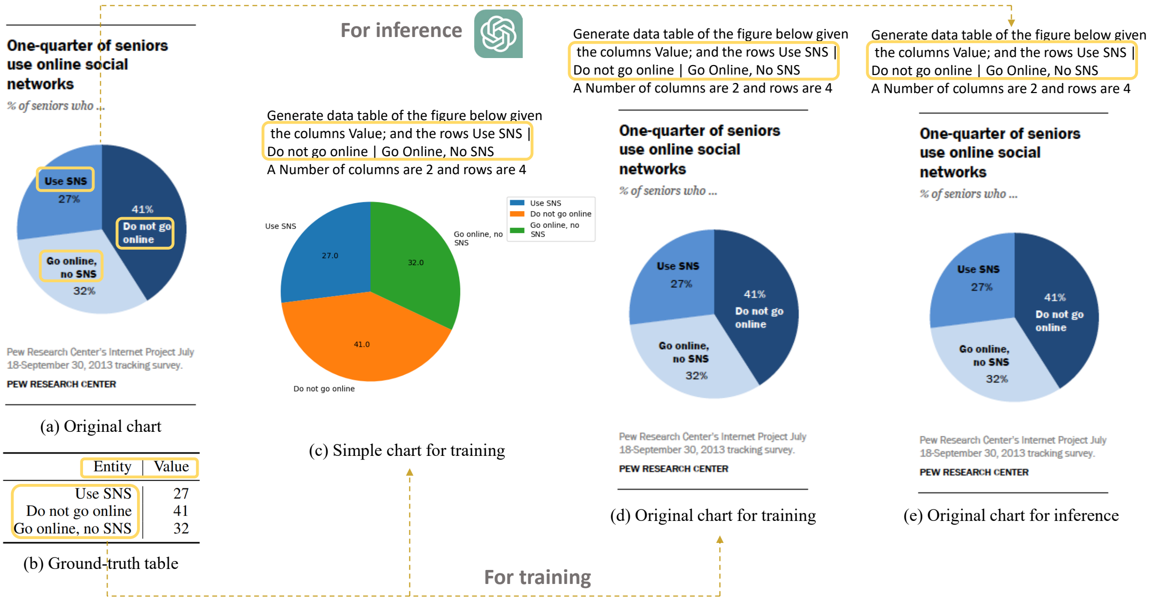
Fig. 7 presents a detailed example of our row-column rendering process. For a simple chart and an original chart used in the training stage, we extract rows and columns from their associated table and directly render them onto the charts since the ground-truth table is provided in this stage. However, for the original chart used in the inference stage, since we cannot access the ground-truth table, we use an LMM to extract rows and columns from the input chart and rendered onto the image.

C Human-oriented Prompt for Chart Reasoning

We present instructions in the prompt to mimic the way humans commonly perceive charts. We name this approach *Human-oriented chart instruction*, which is a simple yet effective methodology as shown in the QA results. The detailed prompt is shown in Fig. 8.

In the *Human-oriented chart instruction*, universal instructions applicable to all chart types are provided first. They include components contained in every chart such as chart title, y-axis, x-axis, legend, and data labels, facilitating the LMM to comprehend the chart. Furthermore, we guide the alignment between a chart and the generated table by instructing the system within the prompt to effectively comprehend the positions of columns and rows as well as the outline of the chart, referring to the table generated by SIMPLOT.

Subsequently, instructions are described for each chart type. Here, we describe the instructions designed for bar charts, but those for line charts and pie charts are designed in a similar manner.



Generate data table of the figure below given the columns Value; and the rows Use SNS | Do not go online | Go Online, No SNS
A Number of columns are 2 and rows are 4

(a) Original chart

Entity	Value
Use SNS	27
Do not go online	41
Go online, no SNS	32

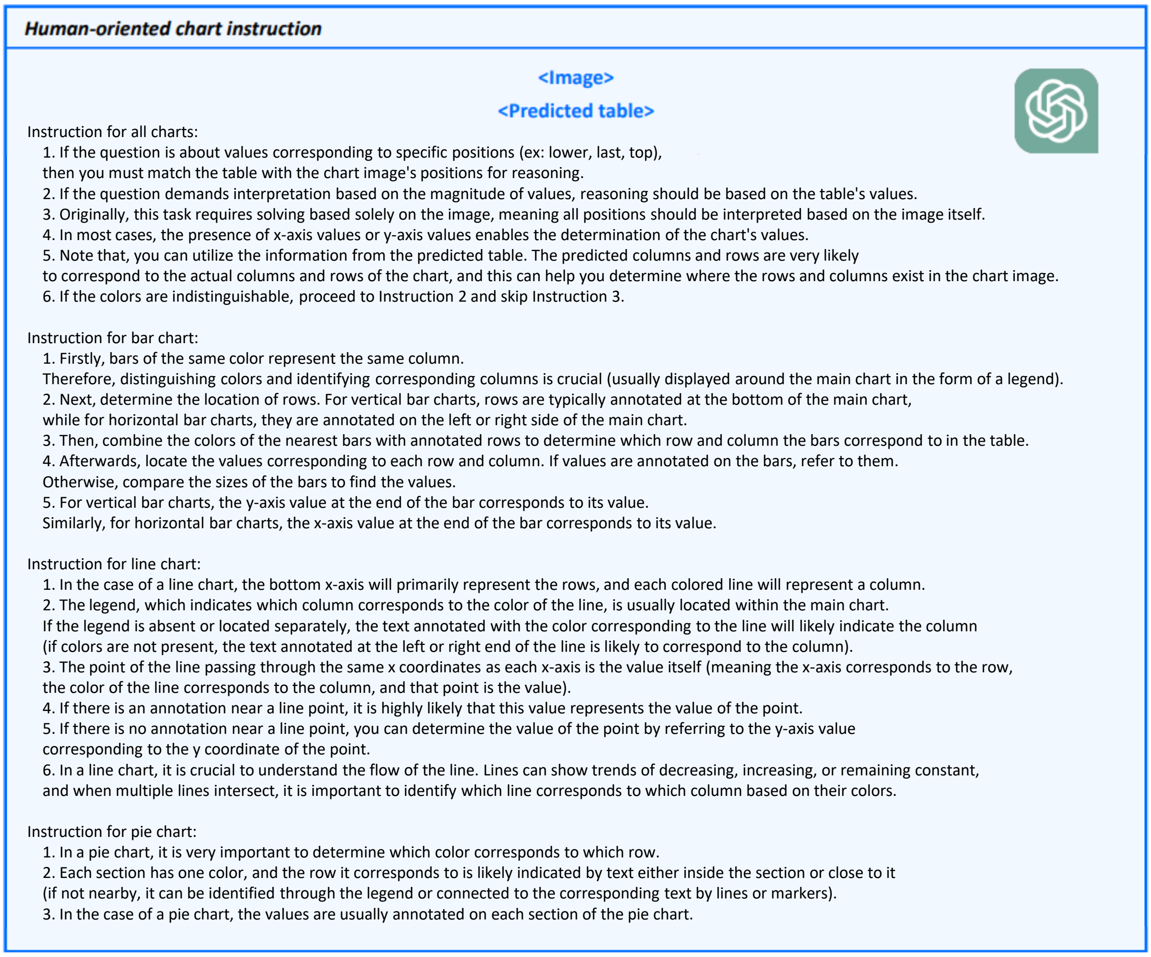
(b) Ground-truth table

(c) Simple chart for training

(d) Original chart for training

(e) Original chart for inference

Figure 7: An example of row-column rendering for an input image in training and inference stage.



Human-oriented chart instruction

<Image>

<Predicted table>

Instruction for all charts:
1. If the question is about values corresponding to specific positions (ex: lower, last, top),
then you must match the table with the chart image's positions for reasoning.
2. If the question demands interpretation based on the magnitude of values, reasoning should be based on the table's values.
3. Originally, this task requires solving based solely on the image, meaning all positions should be interpreted based on the image itself.
4. In most cases, the presence of x-axis values or y-axis values enables the determination of the chart's values.
5. Note that, you can utilize the information from the predicted table. The predicted columns and rows are very likely
to correspond to the actual columns and rows of the chart, and this can help you determine where the rows and columns exist in the chart image.
6. If the colors are indistinguishable, proceed to Instruction 2 and skip Instruction 3.

Instruction for bar chart:
1. Firstly, bars of the same color represent the same column.
Therefore, distinguishing colors and identifying corresponding columns is crucial (usually displayed around the main chart in the form of a legend).
2. Next, determine the location of rows. For vertical bar charts, rows are typically annotated at the bottom of the main chart,
while for horizontal bar charts, they are annotated on the left or right side of the main chart.
3. Then, combine the colors of the nearest bars with annotated rows to determine which row and column the bars correspond to in the table.
4. Afterwards, locate the values corresponding to each row and column. If values are annotated on the bars, refer to them.
Otherwise, compare the sizes of the bars to find the values.
5. For vertical bar charts, the y-axis value at the end of the bar corresponds to its value.
Similarly, for horizontal bar charts, the x-axis value at the end of the bar corresponds to its value.

Instruction for line chart:
1. In the case of a line chart, the bottom x-axis will primarily represent the rows, and each colored line will represent a column.
2. The legend, which indicates which column corresponds to the color of the line, is usually located within the main chart.
If the legend is absent or located separately, the text annotated with the color corresponding to the line will likely indicate the column
(if colors are not present, the text annotated at the left or right end of the line is likely to correspond to the column).
3. The point of the line passing through the same x coordinates as each x-axis is the value itself (meaning the x-axis corresponds to the row,
the color of the line corresponds to the column, and that point is the value).
4. If there is an annotation near a line point, it is highly likely that this value represents the value of the point.
5. If there is no annotation near a line point, you can determine the value of the point by referring to the y-axis value
corresponding to the y coordinate of the point.
6. In a line chart, it is crucial to understand the flow of the line. Lines can show trends of decreasing, increasing, or remaining constant,
and when multiple lines intersect, it is important to identify which line corresponds to which column based on their colors.

Instruction for pie chart:
1. In a pie chart, it is very important to determine which color corresponds to which row.
2. Each section has one color, and the row it corresponds to is likely indicated by text either inside the section or close to it
(if not nearby, it can be identified through the legend or connected to the corresponding text by lines or markers).
3. In the case of a pie chart, the values are usually annotated on each section of the pie chart.

Figure 8: Overall prompt of *Human-oriented chart instruction*.

Bar Chart: Instruction 1. When humans recognize a bar chart, the initial observation would be columns distinguished by their colors. Following this observation, the legend within the image is consulted to identify which color denotes which column. If the colors are indistinguishable, proceed to Instruction 2 and skip Instruction 3.

Bar Chart: Instruction 2. Having identified the columns, humans would proceed to observe which row each bar corresponds to. Specifically, for vertical bar graphs, rows are usually annotated below the main chart, whereas for horizontal bar graphs, annotations on the lateral sides determine the rows.

Bar Chart: Instruction 3. Humans then would match the color of the bar with the annotated row, thereby aligning each bar with its corresponding column and row.

Bar Chart: Instruction 4 & 5. Lastly, humans would figure out the value corresponding to each row and column by referencing the value annotated on the bar, the relative sizes between bars, and the x-axis and y-axis. Notably, it is understood intuitively that the value at the point where the bar terminates on the x-axis or y-axis represents the value of that bar.

We argue that developing a language model that captures the sequential and unconscious reasoning process involved in human chart recognition through *Human-oriented chart instruction* not only mimics human cognition effectively but also ensures the alignment of predicted tables with chart images. This leads to substantial performance enhancements at a lower cost.

D Details regarding Experiments

D.1 Dataset Description

In this paper, we mainly evaluate our proposed method with a widely-used dataset, ChartQA (Masry et al., 2022). It consists of real-world charts containing complex information, which aligns well with our motivation compared to other synthetic datasets such as PlotQA (Methani et al., 2020) and DVQA (Kafle et al., 2018), both of which only include simple charts. Furthermore, ChartQA is comprised of three chart types (i.e., pie, bar, and line), whereas DVQA and PlotQA comprise only one or two types of chart. Therefore, ChartQA is a more challenging dataset and is desirable for evaluating performance on real-world chart reasoning tasks.

For these reasons, we mainly focus on ChartQA dataset in this paper since other datasets are composed only of very simple charts, which are not suitable for our model that is designed to handle real-world charts containing noisy information unnecessary for reasoning. Nevertheless, to demonstrate that SIMPLOT can be applied to any dataset, we conduct additional experiments on the PlotQA dataset in the Appendix E.

ChartQA also includes human-authored and LLM-augmented QA pairs for reasoning. Please note that we do not use QA pairs for training, but train the model to generate tables only from the charts in the training set, then use QA pairs from the test set for measuring reasoning performance.

Detailed statistics of ChartQA dataset are presented in Table 8, and statistics of PlotQA dataset are presented in Table 9. To reduce the training cost and balance the dataset with ChartQA, we stratified and sampled 10% of the images from the PlotQA dataset based on type, which were then used for training and inference. Additionally, for QA pairs, we used one pair per image.

Table 8: Statistics of ChartQA dataset.

split \ type	Pie	Bar	Line	QA pair
Train set	541	15,581	2,195	-
Validation set	48	837	171	-
Test set	78	1,230	211	2,500

Table 9: Statistics of PlotQA dataset.

split \ type	Dot line	Line	Bar	QA pair
Train set	26,010	25,897	105,163	-
Validation set	5,571	5,547	22,541	-
Test set	5,574	5,549	22,534	4,342,514

D.2 Compared Methods

We use a wide range of models capable of performing chart reasoning as baselines, categorizing them into three distinct categories.

The first category includes Vision-Language Pretrained models (VLP) as follows:

- TaPas (Holmgren et al., 2012)
- V-TaPas (Masry et al., 2022)
- T5 (Bujard et al., 1987)
- VL-T5 (Cho et al., 2021)
- PaLI (Chen et al., 2022b)
- Mini-GPT (Zhu et al., 2023)
- LLaVa (Liu et al., 2023)
- GPT-4V (Achiam et al., 2023)

The second category consists of supervised models, including followings:

- ChartQA (Masry et al., 2022)
- ChartT5 (Zhou et al., 2023)
- Pix2Struct (Lee et al., 2023)
- MatCha (Liu et al., 2022b)
- Unichart (Masry et al., 2023)
- ChartLlama (Han et al., 2023)
- ChartAssistant (Meng et al., 2024)
- ChartInstruct (Masry et al., 2024)

	LLM	Textual info.	Designed prompt
Unichart	✗	✗	✗
Deplot	✓	✗	✗
SIMPLOT	✓	✓	✓

Table 10: Comparison of models based on their utilization of LLM, textual information, and designed prompt.

The third category consists of table-based reasoning models, including Deplot (Liu et al., 2022a) and Unichart (Masry et al., 2023). Deplot converts charts to tables similar to SIMPLOT and employs prompt-based methodologies such as Chain-of-Thought (CoT) (Wei et al., 2022), Program-of-Thought (PoT) (Chen et al., 2022a) alongside using an LLM for question reasoning. Although Unichart is trained to perform QA directly from charts, it is also pre-trained to generate tables. In this paper, we use this variant of Unichart to extract tables and utilize LLM to fairly compare with Deplot in the same setting.

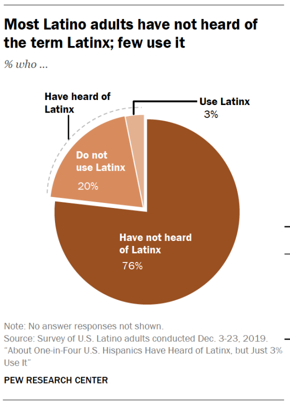
Regardless of our application of Unichart in the same setting as Deplot, the original Unichart requires direct training for each downstream task. It has limitations in terms of adaptability and may exhibit suboptimal performance. Deplot and SIMPLOT achieve performance gains and enhanced adaptability by performing inference through an LLM after chart-to-table extraction. Additionally, SIMPLOT significantly improves the accuracy of table extraction and downstream tasks by leveraging textual information and utilizing prompts specifically designed for charts. The differences among the aforementioned models can be referred to Table 10.

As the existing chart reasoning models (Liu et al., 2022a; Masry et al., 2023, 2024; Meng et al., 2024; Liu et al., 2022b; Masry et al., 2022; Lee et al., 2023) require expensive computational time, they have reported the performance after a single run, which makes it difficult to conduct statistical comparison. Following the previous works, we only report the performance after a single run. However, we claim that our results improve the performance of chart question answering with significant margin.

D.3 Details regarding Proposed Metric

As illustrated in Fig. 9, columns of extracted table from SIMPLOT (e.g., "Percentage (%)" in the figure) often carry the same meaning as the ground truth (e.g., "Value" in the figure) but have different Levenshtein distances. Additionally, as shown in Fig.10, in cases where columns are not explicitly stated in the image, an SIMPLOT utilizing LMM without further fine-tuning generates arbitrary columns. We confidently assert that there are unequivocally no problems associated with chart analysis when employing an LMM in both scenarios. However, given that RMS is composed of the product of Levenshtein distance (from columns and rows) and relative distance (from values), this leads to an apparent degradation in performance of chart-to-table task even when there are no issues with QA. Consequently, we construct a Levenshtein distance matrix by concatenating column and row predictions with the ground truth, followed by applying minimal cost matching to consider only the relative distance of each pair. We adopt this Relative Distance (RD) as our main metric. This approach allows us to accurately assess performance in value mapping.

We follow Deplot (Liu et al., 2022a) to define RD as in the following equation, with most formulas extracted from this paper. Following Liu et al. (2022a), we use basic concept of Related Mapping Similarity (RMS), regarding tables as unordered sets of mappings from row and column headers (r, c) to a single value v. $p_i = (p_i^r, p_i^c, p_i^v)$ and $t_j = (t_j^r, t_j^c, t_j^v)$ indicates each entity for the predicted table $\mathcal{P} = \{p_i\}_{1 \leq i \leq N}$ and the ground truth table $\mathcal{T} = \{t_j\}_{1 \leq j \leq M}$, respectively. Utilizing *Normalized Levenshtein Distance* (NL_τ) (Biten et al., 2019), we denote the distance between two keys p_i and t_j as $NL_\tau(p^r||p^c, t^r||t^c)$ where $||$ indicates concatenation of string. The distance between values of table is computed with relative distance



(a) Input chart

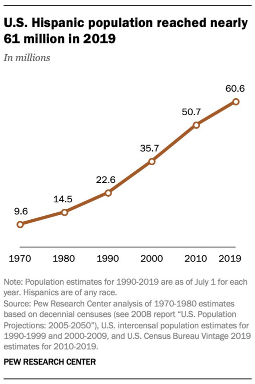
Entity	Value
Have not hearf of Latinx	76
Have heard of Latinx	23
Do not use Latinx	20
Use Latinx	3

(b) Ground-truth table

Entity	Percentage (%)
Have not hearf of Latinx	76
Have heard of Latinx	23
Do not use Latinx	20
Use Latinx	3

(c) Extracted table from SIMPLOT

Figure 9: An example of an extracted table that poses no issues for reasoning despite having a different column name from that in the ground-truth table.



(a) Input chart

Year	Values
1970	9.6
1980	14.5
1990	22.6
2000	35.7
2010	50.7
2019	60.6

(b) Ground-truth table

Year	U.S. Hispanic population (in millions)
1970	9.6
1980	14.5
1990	22.6
2000	35.7
2010	50.7
2019	60.6

(c) Extracted table from SIMPLOT

Figure 10: An example where reasoning remains unaffected, yet a column extracted by SIMPLOT differs significantly due to the lack of clear column description in the image.

$D_\theta(p^v, t^v) = min(1, ||p^v - t^v||/||t^v||)$ and distances larger than θ are set to the maximum of 1. Incorporating these two distances, we obtain the similarity between two entities in a mapping $D_{\tau,\theta}(p, t)$ as $(1 - NL_\tau(p^r||p^c, t^r||t^c))(1 - D_\theta(p^v, t^v))$.

For RMS, Liu et al. (2022a) computes the pairwise similarity in $\mathcal{P}$ and $\mathcal{T}$ using the cost function $(1 - NL_\tau(p^r||p^c, t^r||t^c))$, obtaining a similarity matrix in shape of $N \times M$ and identifying the minimal cost matching $\mathbf{X} \in \mathbb{R}^{N \times M}$ between the keys (in the binary matrix form).

While precision and recall is obtained from two mappings in Deplot (Liu et al., 2022a), we only use distance of the numeric entries to compute RD:

$$RD_{precision} = 1 - \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{X}_{ij}(D_\theta(p^v, t^v))}{N}, \quad (5)$$

$$RD_{recall} = 1 - \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{X}_{ij}(D_\theta(p^v, t^v))}{M}. \quad (6)$$

Similar to Deplot, we compute RD_{F1} as the harmonic mean of the precision and recall.

E Additional Experiments with PlotQA

To demonstrate that our method can be applied to a variety of data beyond the ChartQA dataset, we present additional experimental results on the PlotQA dataset in Table 11. Our proposed method exhibits superior performance across various chart types, indicating that it can be applied to any chart type. However, unlike ChartQA, which consists of real-world charts, PlotQA is composed only of very simple charts, so the performance differences between SIMPLOT and other methods are not as significant as in the result on the ChartQA dataset. We emphasize once again that SIMPLOT is designed to handle real-world charts containing unnecessary and noisy information for chart question answering.

Table 11: Chart question answering performance (RA) on the PlotQA dataset.

Models	Dot line	Line	Bar	Overall
GPT-4V	50.53	58.84	53.85	54.11
Unichart	58.78	53.26	60.10	58.74
Deplot	**66.66**	55.59	61.73	61.53
SIMPLOT	60.93	**65.57**	**73.84**	**70.32**

F Effectiveness of Utilizing Image with Extracted Table

Using tables for chart reasoning allows for performing various downstream tasks (such as chart summarization, chartQA, etc.). However, there is an inherent limitation of using tables for chart reasoning, i.e., incorrect extraction of the table leads to failure in all subsequent downstream tasks. Taking this limitation into account, unlike Deplot that solely relies on tables for reasoning, SIMPLOT simultaneously considers both the extracted table and the image itself to compensate for the incorrectly extracted table.

To further verify the effectiveness of using an image alongside the generated table, we conduct additional experiments. Table 12 proves that using images along with tables increases the success rate of QA compared to using only tables. We assume both Deplot and SIMPLOT failed to accurately extract the table in cases where the Relative Distance (RD) was below 0.7 (i.e., when the SIMPLOT does not benefit from utilizing simple charts) and the difference between the two RDs was within 0.1, and there are 155 such cases. Among these cases, even though we provide images to Deplot and removed the prompt from SIMPLOT for a fair comparison, while Deplot failed in QA, SIMPLOT succeeded in 48 cases, which is more than 30% additional correct answers. Conversely, when SIMPLOT failed in QA, Deplot succeeded in only 7 cases, which is less than 5%. This demonstrates that utilizing both plot and table can effectively address the limitation of solely relying on tables for reasoning.

Table 12: The number of times QA succeeded or failed when both methods failed to extract the table properly.

SIMPLOT Deplot	Success	Fail
Success	-	7
Fail	48	-

G Hyperparameter Analysis

We conduct a hyperparameter anaylsis by setting various values for the balancing parameter λ and analyzing the result changes in RD. As illustrated in Fig. 11, the changes in RD values due to variations in λ are minimal. This indicated that our model is robust and performs consistently well regardless of the λ value.

H Effectiveness of Negative Sample

For more effective learning, we can generate negative samples at no additional cost and use them for training. Charts of the same type (e.g., line charts) tend to have a similar format even if they contain different content, so the representations obtained from the model will likely be similar. Since the value information within the chart is critical, we utilize negative samples that have a format similar to positive samples but contain different values, allowing the model to capture finer differences. In other words, the purpose of utilizing negative samples is to more precisely discriminate between data that may be mapped closely in the representation space because the charts look similar. The effect of using negative samples together is shown in Table 13. Although the difference in table extraction performance may appear minor, even a slight improvement in extracting values more accurately is important in the process of converting charts into tables.

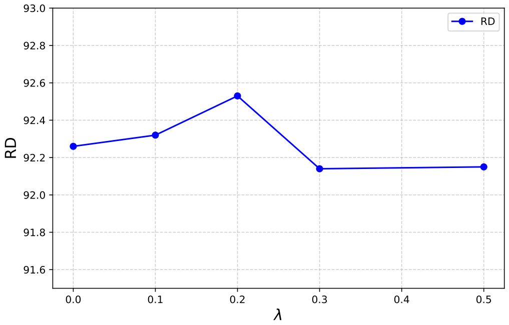


Figure 11: Table extraction performance (RD_{F1}) with varying λ.

Positive	Negative	RD_{F1}
✓	✗	92.18
✓	✓	92.32

Table 13: Ablation studies on utilizing negative sample SIMPLOT for chart-to-table extraction.

I More Case Studies

Here, we present more case studies. We confirm that rendering columns and rows of a table onto the image is beneficial for extracting an accurate table. In Fig. 13, Deplot (Liu et al., 2022a) omits a relevant row (i.e., "Montenegro"), redundantly extracts several rows, and even generates entirely irrelevant rows such as "CC BY," which would result in a poor performance in reasoning tasks. In contrast, SIMPLOT accurately generates a table by extracting only the relevant rows, ensuring accurate reasoning.

Fig. 14 and 15 show that in both pie charts and bar charts, Deplot (Liu et al., 2022a) fails to capture relevant rows, generates inaccurate rows, thus failing to create values. On the other hand, SIMPLOT consistently produces accurate tables. This directly proves the generality of SIMPLOT across various chart types.

Furthermore, Fig. 17 demonstrates that our *Human-oriented chart instruction* works well by imitating the way of human interpretation for charts. Without the proposed prompt, an LLM fails to detect the exact location of the line component "Favor", leading to a wrong answer. However, utilizing *Human-oriented chart instruction*, an LMM can precisely locate the necessary component, resulting in a correct answer.

J Evaluations on more Challenging Questions

While conducting case studies, we observed that Deplot and Simplot often exhibited similar QA performance, despite Deplot failing to accurately extract tables that SIMPLOT successfully extract. *We attribute this mainly to the simplicity of the questions in the ChartQA dataset.* For example, in Fig. 15, although Deplot fails to accurately extract the table, if the question happens to be about 'Dem/Lean Dem', which is not related to the inaccurate part of the extracted table, Deplot can still generate the correct answer. This proves that the performance of our method can be further differentiated with more complex questions.

Hence, we generated more challenging questions, and evaluated models in Section 3.2. Fig. 16 shows an example of challenging question, where Deplot fails to answer correctly due to its poorly extracted table, while SIMPLOT generates a precise answer based on its accurately extracted table. The question is *"What is the sum of the birth rate in China in 1955 and the death rate in China in 1965?"*, which requires more than one row and column to answer. In this example, we verify that extracting accurate tables is crucial for solving challenging questions, and that the ChartQA dataset is insufficient to precisely evaluate the reasoning models.

K Error Analysis

Here we present examples of SIMPLOT failures, their reasons, and possible solutions. In Fig. 18 (c), it can be observed that the extracted table from the proposed method contains inaccurate rows. This is attributed to a line break between the items "Exposure to false or incorrect information" and "Losing the ability to Communicate face-to-face" in the input chart, leading to the misinterpretation of each item as two separate entities during row extraction. To address such issues, one possible solution would be to refine the rows and columns by considering their contextual information before rendering.

In addition, we find some examples where the LMM lacks specific abilities, such as color detection. In Fig. 19, there is a question asking about the color of "Jamaica" with the correct answer being orange, but the model recognizes it as red and answers incorrectly. One possible solution would be to broaden the range of similar colors for the model to answer together.

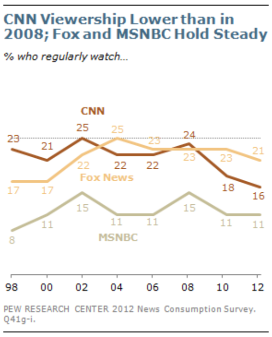
L Responsible NLP Research Checklist

Hardware.

We train SIMPLOT using a NVIDIA RTX A6000. The training time for phase 1 (Section 2.2.1) and phase 2 (Section 2.2.2) are about 2 hours and 4 hours per epoch, respectively. We train phase 1 for 7 epochs and phase 2 for 9 epochs.

Parameters.

SIMPLOT has 374M parameters, and the exact number of parameters in GPT-4 has not been publicly reported. For LMM prompting, we use temperature of 0.1.



Question: What is the % who regularly watch CNN in 02? Correct Answer: 25

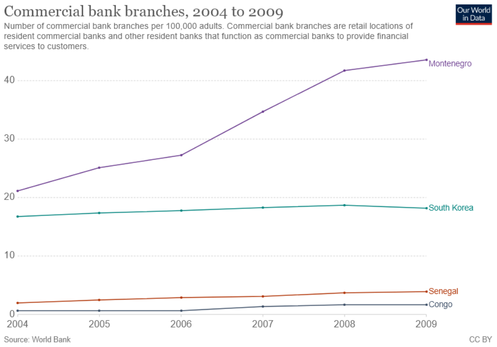
SIMPLOT w/o prompt

Explanation: The percentage of people who regularly watch CNN in the year ’02 is found in the chart under the CNN column at the point where the year ’02 is marked. The chart shows that the value is 21%.
Answer: 21

SIMPLOT

Explanation: To answer the question, we must refer to the line chart. The x-axis represents the years, which correspond to the rows in the table. Each line represents a different news network, which corresponds to the columns in the table. The value for CNN in a specific year can be found by locating the point on the CNN line (red) that aligns with the year on the x-axis and then reading the value from the y-axis.
The chart shows the percentage of people who regularly watch CNN, Fox News, and MSNBC over various years. The red line represents CNN. In 2002, the red line is at the 25% mark on the y-axis.
Answer: 25

Figure 12: Case study. Interpreting charts similar to how humans do using *Human-oriented chart instruction*.



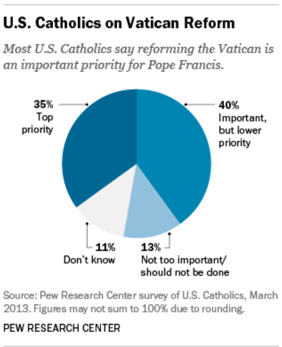
(a) Input chart

Entity	Commercial bank branches, 2004 to 2009
South Korea	16.86
Congo	4.16
Senegal	3.12
Congo	3.7
Senegal	4.32
Como	4.0
CC BY	+ 1 missing row 4.19

Entity	2004	2005	2006	2007	2008	2009
Montenegro	21.4	25.5	27.0	34.5	42.1	43.2
South Korea	16.8	17.0	17.9	18.3	18.7	18.2
Senegal	2.0	2.1	2.2	2.3	2.7	2.8
Congo	0.3	0.3	0.3	0.3	0.3	0.3

(b) Extracted table from Deplot (upper) & SIMPLOT (lower)

Figure 13: Case study. Extract precise rows without unnecessary information in the chart.



(a) Input chart

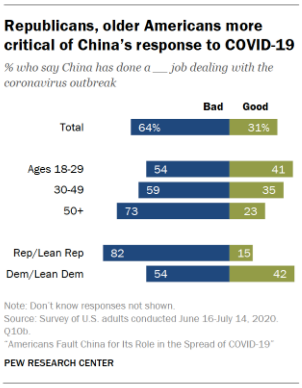
Entity	Value
Don’t know	11
Not too important/ should not be done	13
Don’t know	11
Top priority	35
Top candidate	40

+ 1 missing row

Entity	Value
Top priority	35
Important, but lower priority	40
Don’t know	11
Not too important/ should not be done	13

(b) Extracted table from Deplot (upper) & SIMPLOT (lower)

Figure 14: Case study. Extract all necessary rows in the pie chart.



(a) Input chart

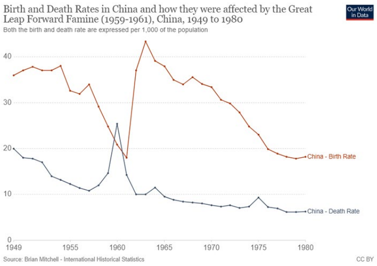
Entity	Bad	Good
Dem/Lean Dem	54	42
Rep/Lean Rep	82	15.0
50+	73	23.0
Ages 18-29	54	nan
Total	64	31.0

+ 1 missing row

Entity	Bad	Good
Dem/Lean Dem	54	42
Rep/Lean Rep	82	15
50+	73	23
30-49	59	35
Ages 18-29	54	41
Total	64	31

(b) Extracted table from Deplot(upper) & SIMPLOT(lower)

Figure 15: Case study. Extract all necessary rows and precise value in the bar chart.



(a) Input chart

Entity	1949	1955	1960	1965	1970	1975	1980
Birth Rate	37.8	nan	nan	nan	nan	nan	18.4
Death Rate	17.2	nan	nan	nan	nan	nan	6.5

Entity	1949	1955	1960	1965	1970	1975	1980
Birth Rate	35.9	32.5	25.3	37.7	33.5	24.3	18.2
Death Rate	18.2	12.2	14.8	9.6	7.4	9.4	6.2

(b) Extracted table from Deplot (upper) & SIMPLOT (lower)

Question: What is the sum of the birth rate in China in 1955 and the death rate in China in 1965?? Correct Answer: 42.1

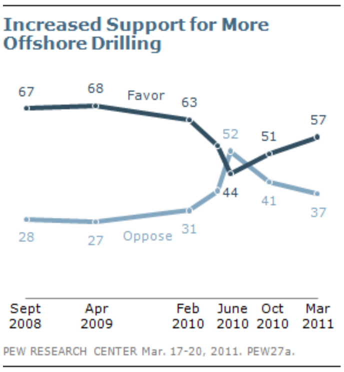
Deplot + img. + prompt

Explanation: For a line chart, the x-axis typically represents the rows (time periods in this case), and each line represents a different column (birth rate and death rate). The value of the line at a given year can be determined by the y-axis value that the line intersects. Since the question asks for specific years, we locate the points on the lines that correspond to the years 1955 and 1965 and sum the values of the birth rate in 1955 and the death rate in 1965.
Answer: 47

SIMPLOT

Explanation : To find the sum of the birth rate in China in 1955 and the death rate in China in 1965, we need to identify the values from the line chart. The x-axis represents the years, and the two lines represent the birth and death rates, respectively. We locate the year 1955 on the x-axis and find the corresponding point on the birth rate line, then do the same for the year 1965 on the death rate line. The y-axis provides the values for these points. Since the question is about specific years, we refer to the predicted table for the exact values for those years.
Answer: 42.1

Figure 16: Case study. Effectiveness of precise extracted table for challenging question.



Question: Which group favor or oppose has the highest value in the year Sept 2008? Correct Answer: Favor

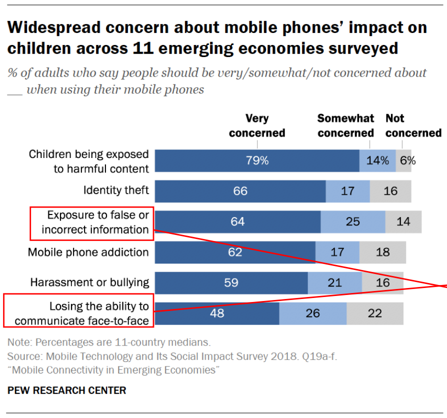
SIMPLOT w/o prompt

Explanation: According to the table, in Sept 2008, the value for 'Favor' is 0 and the value for 'Oppose' is 28. Therefore, the 'Oppose' group has the highest value in the year Sept 2008.
Answer: Oppose

SIMPLOT

Explanation: For a line chart, the x-axis typically represents the rows (time periods in this case), and each line represents a different category or column. The points on the lines correspond to the values for each category at a given time. To find the highest value for a specific time period, you need to look at the points on the chart that align with that time period on the x-axis and compare the y-axis values of those points. Based on the line chart, in Sept 2008, the point on the 'Favor' line is higher than the point on the 'Oppose' line. The 'Favor' line is at approximately 67, and the 'Oppose' line is at 28, as indicated by the annotations on the chart.
Answer: Favor

Figure 17: Case study. Interpreting charts similar to how humans do using *Human-oriented chart instruction*.



(a) Input chart

Entity	Very concerned	Somewhat concerned	Not concerned
Losing the ability to communicate face-to-face	48	26	22
Harassment or bullying	59	21	16
Mobile phone addiction	62	17	18
Identity theft	66	17	16
Children being exposed to harmful content	79	14	6

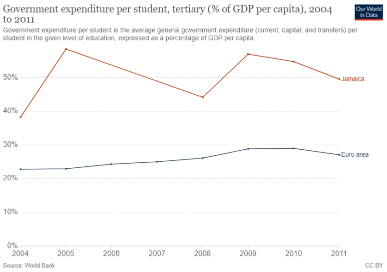
+ 1 missing row

(b) Extracted table from Deplot

Entity	Very concerned	Somewhat concerned	Not concerned
Communicate face-to-face	48	26	22
Losing the ability to	48	26	22
Harassment or bullying	59	21	16
Mobile phone addiction	62	17	18
Incorrect information	64	25	14
Exposure to false or	64	25	14
Identity theft	66	17	16
Children being exposed to harmful content	79	14	6

(c) Extracted table from proposed method

Figure 18: Error Analysis. Failure in row / column rendering.



Question: What's the color of the line of Jamaica? Correct Answer: Orange

SIMPLOT

Explanation: To determine the color of the line representing Jamaica in a line chart, one should look for the legend or any text annotations that are connected to the line by color. The legend or annotations will indicate which color corresponds to Jamaica. The line representing Jamaica in the chart is colored red, as indicated by the legend or text annotation connected to the red line.
Answer: Red

Figure 19: Error Analysis. Failure in detecting a color in the chart.