Assessing Y-Axis Influence: Bias in Multimodal Language Models on Chart-to-Table Translation

Seok Hwan Song, Azher Ahmed Efat, Wallapak Tavanapong
Department of Computer Science, Iowa State University, Ames, Iowa, USA
{song92, efat, tavanapo}@iastate.edu

Abstract

Chart-to-table translation converts chart images into structured tabular data. Accurate translation is crucial for Multimodal Language Model (MLM) to answer complex queries. We observe imbalances in the number of images across different aspects of the y-axis information in public chart datasets. Such imbalances can introduce unintended biases, causing uneven MLM performance. Previous works have not systematically examined these biases. To address this gap, we propose a new framework, **FairChart2Table**, for analyzing y-axis-related bias on five state-of-the-art models.

Key Findings: (1) There are significant y-axis biases related to the digit length of the major tick values, the number of major ticks, the range of values, and the tick value format (e.g., abbreviation or scientific format). (2) The number of legends/entities in chart images impacts MLM performance. (3) Prompting MLM with y-axis information can significantly enhance the performance for some MLMs.

1 Introduction

Multimodal Language Models (MLMs) have been evaluated on diverse tasks in chart image understanding, such as chart image summarization (Islam et al., 2024), question answering (QA) (Masry et al., 2022), fact checking (Akhtar et al., 2023), and chart-to-table translation—translation of data in chart images to corresponding tabular data (Liu et al., 2023a).

Biases in MLM performance on QA have been studied. These biases include chart complexity measured by the number of entities (data series), color scheme, font size, grid lines, legend position, scale of the figure, logarithmic scale in the y-axis ticks, and the x-axis tick orientation (Mukhopadhyay et al., 2024), along with chart image resolutions (Pramanick et al., 2024).

Dataset	Digit Length Range	CV (Std./Mean*)
PlotQA	0–16	**1.72** (15040.9/8769.6)
ChartQA	0–11	**1.40** (2441.0/1738.1)
FairChart2Table (Ours - Part A)	0–16	**0.00** (0.0/180.0)

Table 1: Imbalance in datasets by digit length; CV stands for coefficient of variation (Pearson, 1896). *Mean and Std. are in terms of #images per digit length.

This paper focuses on unexplored biases related to the y-axis information in chart-to-table translation tasks. Accurate chart-to-table translation enables more accurate reasoning than direct image-based QA (Kim et al., 2025), and improves interpretability in complex reasoning with the extracted tabular data. The y-axis has different information important for accurate value extraction, such as the minimum and maximum values on the y-axis, the number of major and minor ticks, and the tick value format—the format of the number placed next to the tick marks, such as a scientific format, numbers with commas and abbreviations like **K**, **M**, **B**, **T** for a thousand, a million, a billion, and a trillion, respectively.

To observe a potential bias related to the y-axis, we analyzed two well-known chart datasets, PlotQA (Methani et al., 2020) and ChartQA (Masry et al., 2022). We characterize the y-axis maximum value by the number of digits in its integer part, referred to hereafter as *digit length*. For example, a number in the range of (-1, 1) has zero digit length; numbers in the range of [1,10) or [-1, -10) have the digit length of 1. The longer the digit length, the larger the positive or negative value. Table 1 shows the imbalance in the number of images across digit lengths in the existing datasets and our proposed dataset without such biases (CV=0).

This imbalance, along with other imbalances

Findings of the Association for Computational Linguistics: ACL 2026, pages 35412–35430

related to the y-axis information, can lead to unintended biases, ultimately impacting MLM performance in chart-related tasks. Consequently, models may excel on chart images with specific y-axis characteristics but struggle to generalize to others.

To our knowledge, the influence of the y-axis information on MLM performance for chart-to-table translation has not been systematically investigated, and no benchmark for the evaluation exists. We propose a new **Bias Controlled Chart-to-Table (FairChart2Table) Framework**. The framework includes (1) dataset generation methods designed to eliminate confounding biases on the y-axis and (2) new performance metrics, considering visually noticeable errors more seriously than small fluctuations. *To quantify these visible errors, we define Tick-Based Error (TBE) as the ratio of the absolute difference between the ground truth and the predicted value to the minor tick interval.* Using our framework, we investigate the following research questions for chart-to-table translation.

1. (RQ1) How does the y-axis information affect MLM performance?
2. (RQ2) What other factors of chart images (e.g., chart styles, crossing lines) impact MLM performance?
3. (RQ3) Does prompting MLMs with explicit y-axis information help improve MLM performance?

Our contributions are summarized below.

Contribution #1: The aforementioned research questions that have not been explored.

Contribution #2: The FairChart2Table framework, consisting of the methodology to generate the benchmark dataset for the above research questions and the new TBE-based performance metrics. We will share our code and dataset publicly[1]. The benchmark can be used to test biases in other MLMs.

Contribution #3: New key findings from evaluations of five models on chart-to-table translation. These models include two closed-source, one open-source, and three open-source fully supervised models. Some key findings are as follows. (1) Y-axis related biases exist across multiple configurations, including the number of major ticks, the numerical range of the axis, and the tick value format. These factors can significantly influence model performance. (2) Models tend to get confused about the position of numerical values on the same x-axis tick when the number of entities, corresponding to the number of distinct elements in the chart legend, increases. In contrast, the chart type appears to have a minor influence. (3) Prompting MLMs with the y-axis tick values for chart-to-table translation can enhance model performance.

[1]Code and benchmark: https://github.com/NRT-D4/FairChart2Table

2 Related Work

2.1 Chart Benchmarks

Chart Datasets: DVQA (Kafle et al., 2018) and FigureQA (Kahou et al., 2017) are among the earliest datasets introduced for chart-based question answering. PlotQA (Methani et al., 2020) and ChartQA (Masry et al., 2022) are widely used for chart-based question answering. PlotQA has synthetic plots created from real-world data, and the questions were generated using templates. ChartQA consists of human-written questions as well as automatically generated questions from human-written summaries. LEAF-QA (Chaudhry et al., 2020) is a large dataset with 2 million question-answer pairs. Recent work includes datasets on multiple chart images (Zhu et al., 2025; Liu et al., 2024; Kantharaj et al., 2022).

Performance Metrics for Chart-to-Table Translation: Relative Mapping Similarity F1 (RMS_{F1}), proposed by Liu et al. (2023a) is widely used (Zhang et al., 2024; Kim et al., 2025; Masry et al., 2023; Meng et al., 2024). This metric assesses the alignment between the predicted and ground truth tables, considering both textual and numeric elements and the table structure (column and row headers). RMS_{F1} calculates the F1 score using the combined distance function, incorporating the normalized Levenshtein distance for textual mismatch, and the relative numeric distance for numerical values. Minimal cost matching is used to match the values between the target and predicted tables.

Models: Recent models are categorized into three groups. 1) Specialized models such as DePlot (Liu et al., 2023a) and Simplot (Kim et al., 2025) were designed for chart-to-table translation by leveraging encoder-decoder architectures tailored to visual inputs. 2) Matcha (Liu et al., 2023b), UniChart (Masry et al., 2023), and TinyChart (Zhang et al., 2024) are supervised models trained on general chart understanding tasks. 3) General-

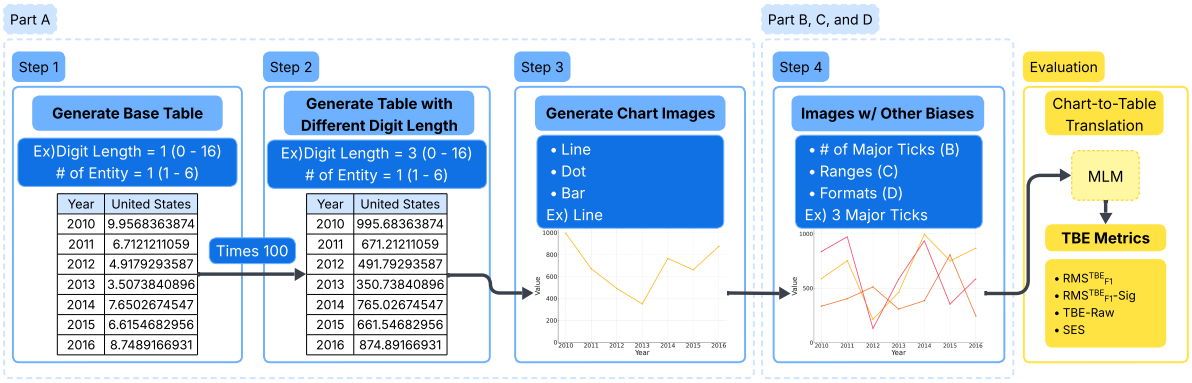

Figure 1: FairChart2Table Framework. Step 1 generates 10 base tables for each entity count (1-6) with a digit length of 1. Step 2 derives 1,020 tables by scaling the values to the target digit length (0-16) using powers of ten. Step 3 creates 3,060 chart images representing three chart types based on the generated tables. Step 4 chooses only the tables with three entities from Part A, and produces additional images by varying the number of major ticks, ranges, and formats. Last, evaluate MLMs for chart-to-table translation using our TBE-based metrics.

purpose MLMs like GPT-based multimodal models such as GPT-4o offering promising results on chart-related tasks (Zhu et al., 2025; Islam et al., 2024; Mukhopadhyay et al., 2024). Instruction tuning was shown to enhance performance on chart reasoning tasks (Masry et al., 2025, 2024). Gupta et al. (2024) pretrained existing models on simple tasks, structural and visual knowledge, data extraction, and numeric question answering.

2.2 Biases in MLMs for Chart-Based Tasks

Aspect	Factor Manipulation
Complexity	Single entity and $\geq$2 entities
Color scheme	palettes, gradients, and hues
Font size	Small and Big
Grid lines	Presence / density
Legend position	Placement
Figure scale	Overall size
Tick scale (log)	Linear and log
Tick orientation	Horizontal and angled/vertical
Image resolution	pixel count

Table 2: Existing work on bias analysis of MLM performance (Mukhopadhyay et al., 2024; Pramanick et al., 2024)

Table 2 summarizes existing work on bias analysis. Bias from digit length was not considered. Our work also investigates the impact of varying the number of entities from 1 to 6, more than the number studied in (Mukhopadhyay et al., 2024). Unlike the work by Mukhopadhyay et al. (2024), which evaluates model performance solely on a log scale in ticks, we consider three other types of tick value number format (comma, scientific notation, and abbreviation types).

3 Proposed Bias Controlled Chart-to-table (FairChart2Table) Framework

Current datasets contain diverse components, such as chart types, styles, and content, which can introduce various forms of bias. As a result, it is challenging to use them to effectively identify and evaluate biases in the chart-to-table translation task. Additionally, existing evaluation metrics are insufficient to capture easily noticeable visual errors, thereby limiting the reliability of current assessments. To address these issues, we propose **Bias Controlled Chart-to-Table (FairChart2Table) Framework** with a new method to generate a new dataset free from bias related to the y-axis information and new evaluation metrics. The proposed framework enables investigations of our research questions and can be used to evaluate the y-axis biases in other MLMs.

3.1 Benchmark Construction

Figure 1 illustrates the process to generate our benchmark. To avoid other unintended biases, we generated all chart images using the Bokeh plotting library as in PlotQA (Methani et al., 2020), one of the most widely adopted datasets for training state-of-the-art models (Liu et al., 2023a; Masry et al., 2023; Zhang et al., 2024). Additionally, chart elements, including titles, units, labels, font styles, font sizes, legend locations, x-axis information, and color combinations, are standardized.

The dataset has four subsets (A-D), each focusing on a distinct y-axis-related bias: digit length, number of entities, number of major ticks, value ranges, and tick value formats. Each dataset is designed to isolate a single bias, ensuring no confounding factors.

Part A for the evaluation of digit length bias: We generated 10 tables for each entity count from 1 to 6, using values within a single-digit length, 0 to 10. These values in the tables are scaled by powers of ten to the target digit length from 0 to 16. From these 1,020 (10*6*17) tables, we generated one line, one dot, and one bar chart. The y-axis has 6 major ticks, with the tick values in plain numerical format, and the intersection of the x-axis and y-axis at the origin. This dataset enables a fair evaluation of model performance across different digit lengths and entity counts. Part A has a total of 3,060 chart images.

Parts B-D: To eliminate the number of entities as a confounding factor, we chose only the tables with three entities from Part A, resulting in 170 tables to use as the base for generating the remaining parts. The specific details for each are below.

Part B for the evaluation of major tick bias has 170 chart images with 3 major ticks and 170 chart images with 11 major ticks for performance comparison with the chart images with 6 major ticks and three entities from Part A.

Part C for the evaluation of y-axis range bias has 3*170 chart images with three different ranges of y-axis values, varying from those of the 170 tables in Part A as follows.

- Positive minimum tick value (Pos): All values are shifted upward by adding three times the major tick interval, ensuring the minimum tick value is positive. The corresponding chart has a positive displaced y-axis origin.
- Negative minimum tick value (Neg): All values are shifted downward by subtracting three times the major tick interval, ensuring the minimum tick value is negative. The corresponding chart has a negative displaced y-axis origin.
- Extended range (Ext): The data values remain unchanged, but the maximum tick value is double, visually pushing the data values to the bottom of the chart.

We do not characterize the minimum tick values by the digit length since we did not observe a performance impact due to the digit length of the negative values.

Part D for the evaluation of number format bias has chart images with three variants of the y-axis label formats. Numbers can be represented with commas (e.g., 7,000), abbreviations (K, M, B, and T), and a scientific format (e.g., 7.00e + 6).

The total of chart images for this dataset is 7,140 images (3060 + (2*170 + 2*3*170)*3). Appendix A.5 shows chart image examples.

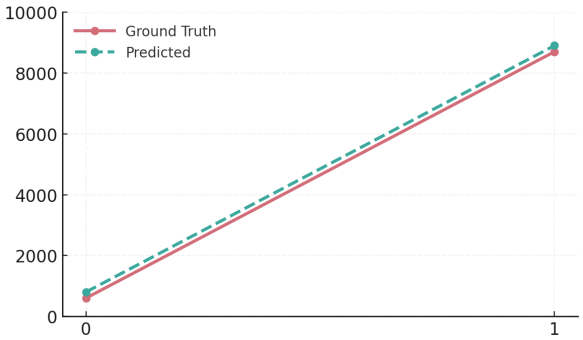

Figure 2: Ground truth and predicted values differ by the same amount, 200 points, at both $x = 0$ and $x = 1$. For these two points, the distance measured by RMS_{F1}^{TBE} is 0.5 in both cases, while RMS_{F1} yields 0.33 at $x = 0$ and 0.02 at $x = 1$. Visually, RMS_{F1}^{TBE} reflects the equal difference between the predicted and the ground truth values better.

3.2 Evaluation Metrics

RMS_{F1} (Liu et al., 2023a) is a widely used metric for table-based evaluation. It first aligns the predicted and ground truth tables using its own matching procedure and then scores the aligned cells with Eq. 1 as the distance.

$$D_{\text{RMS}}(g, p) = min\left(1, \frac{|g - p|}{|g|}\right), \qquad (1)$$

where g and p reflect the ground truth and predicted values of aligned cells. However, the distance function does not capture visual differences well. As illustrated in Figure 2, visually equivalent differences can receive different scores under D_{RMS}, leading to inconsistent scoring for the chart-to-table task.

We introduce and investigate several **Tick-Based Error (TBE)** metrics. TBE in Eq. 2 normalizes the absolute difference between the ground truth g and the predicted value p by a fixed amount $t > 0$ that estimates the minor tick interval. As Figure 2 shows, the distance of RMS_{F1}^{TBE} gives

visually more accurate scores. Our rationale is that an error larger than a minor tick interval is visually noticeable on a chart. Our implementation uses one-fifth of the major tick interval of an input chart to estimate the minor tick interval to ensure that TBE can be computed for all charts, including those without minor ticks.

3.2.1 Proposed RMS_{F1}^{TBE} Metric

The calculation of RMS_{F1}^{TBE} score follows that of the RMS_{F1} score (Liu et al., 2023a) except for two changes. (1) We use the TBE distance in Eq. 2 for the numerical error.

$$D_{\text{TBE}}(g, p) = min\left(1, \frac{|g-p|}{t}\right) \tag{2}$$

(2) Once the column and row headers of the truth and predicted table from chart-to-table translation are matched, no further penalties for headers are applied to the associated value cells. As Kim et al. (2025) noted, Levenshtein distances used in RMS_{F1} for column and row headers may not be close for synonym words. Moreover, MLMs sometimes incorporate information from the titles and labels of a chart into the predicted headers, which do not alter the meaning of the headers but decrease the RMS_{F1} score. Although we still use the matching procedure in RMS, using *Normalized Levenshtein Distance* to match headers between the truth and the predicted tables, we do not repeatedly apply it across cells, reducing over-penalization and improving fairness in the overall error aggregation.

Since our analysis focuses on biases with numerical information, we exclude the header similarity scores from the main results. For completeness, Appendix A.2 presents both RMS_{F1} and RMS_{F1}^{TBE} to demonstrate over-penalization due to the table header. RMS_{F1}^{TBE} values are in the interval [0,1]. The higher the values, the closer the predicted values are to the ground truth values. Appendix A.1 shows the equation for RMS_{F1}^{TBE}.

3.2.2 Proposed RMS_{F1}^{TBE}-Sig and TBE-Raw Metrics

These metrics employ the same mapping procedures as RMS_{F1}^{TBE} but differ in their choice of distance function and aggregation scheme. Specifically, RMS_{F1}^{TBE} employs D_{TBE} for the distance and aggregates through the harmonic mean ($F1$). The distance used in RMS_{F1}^{TBE}-Sig shown in Eq. 3 considers only *significant* deviations—those with the absolute error greater than the value of t, to avoid the dilution effect in tables with many data points.

$$D_{\text{TBE-Sig}}(g, p) = \mathbf{1}\left\{\frac{|g-p|}{t} \geq 1\right\}, \tag{3}$$

where $\mathbf{1}\{c\}$ is an indicator function that returns 1 when the condition c is true; otherwise 0. The metric RMS_{F1}^{TBE}-Sig considers only the cells with significant deviations in the calculation.

The TBE-Raw metric utilizes Eq. 4 to calculate the distance between individual aligned cells in the predicted and truth tables. This distance function does not suppress any error magnitudes as in RMS_{F1}^{TBE}. The metric TBE-Raw is the mean of all cell's $D_{\text{TBE-Raw}}$ from the tables. Therefore, the values are in the range of [0,∞).

$$D_{\text{TBE-Raw}}(g, p) = \frac{|g-p|}{t} \tag{4}$$

3.2.3 Swapping Error Score (SES)

We introduce **Swapping Error Score (SES)** shown in Eq. 5 to measure errors commonly observed when numbers are correctly extracted, but the entities they belong to are swapped. This situation often occurs when MLMs process a chart image with intersecting lines. At a given x-axis value, the extracted y-value of one line (entity) is assigned to an entity of a crossing line and vice versa.

$$\text{SES} = \text{RNSS}_{F1}^{TBE} - \text{RMS}_{F1}^{TBE}, \tag{5}$$

where RNSS_{F1}^{TBE} measures the similarity between the values in the ground truth and the predicted tables. To calculate RNSS_{F1}^{TBE}, we use a minimal cost matching algorithm to minimize relative errors across all value pairs, regardless of the order, as described in (Liu et al., 2023a). However, we make one difference: utilizing D^{TBE} for numerical error calculations and the harmonic mean. We subtract RMS_{F1}^{TBE} to isolate the impact of swapped numbers, offering a clearer view of structural errors beyond the accuracy of the raw value. SES values are in the range of [-1, 1].

None of these metrics, including the original RMS_{F1}, captures the differences in the chart titles.

3.3 Models and Experiments

We selected TinyChart (Zhang et al., 2024) to represent fully supervised models for tasks in chart image understanding, since it performs the best on chart-to-table translation among SOTA models (Zhang et al., 2024). We additionally included

Figure 3: RMS_{F1}^{TBE} performance for bias evaluation by digit length (X-axis); "W/Hint" indicates when the y-axis major tick values were used as part of the prompts to MLMs.

ChartGemma (Masry et al., 2025), since it is, like TinyChart, a model trained on multiple chart understanding tasks. DePlot (Liu et al., 2023a) was selected to represent fully supervised models only for chart-to-table translation, since it performs better than UniChart overall. Simplot was not compared since it requires additional fine-tuning for different chart styles (Kim et al., 2025). For closed-source MLMs, we chose GPT-4o (gpt-4o-2024-05-13) (Hurst et al., 2024) and Gemini-2.0-flash, and chose Pixtral-12B-2409 (Agrawal et al., 2024) to represent an open-source MLM for repeatability of experiments. The default configurations used greedy decoding (i.e., temperature = 0). For each reasoning problem and configuration, a single API call was made to an MLM to facilitate paired statistical tests. The experiments were conducted in late September 2025.

3.4 Statistical Test

Hereafter, we present all metric values as percentages, those ranging from 0 to 1 (RMS_{F1}^{TBE} and RMS_{F1}^{TBE}-Sig), and those ranging from -1 to 1 (SES). We conducted paired two-sided Wilcoxon signed-rank tests comparing the performances of a given model on the same charts under each condition against the baseline, treating each chart as a paired observation. We employed rank-based tests for the metrics with the same bounded ranges, ensuring robustness due to departures from normality and the presence of many outliers.

DL	DePlot	Chart Gemma	Tiny Chart	Pixtral	Gemini	GPT-4o
0	3.06	4.01	0.26	0.38	0.39	3.41
1	0.53	3.79	0.16	0.47	0.37	3.39
2	0.58	3.63	0.16	0.49	0.31	3.49
3	0.67	3.90	0.17	0.57	0.43	3.50
4	1.07	5.70	0.42	0.55	0.45	3.44
5	3.03	6.85	0.35	1.23	0.38	3.63
6	2.44	8.95	0.54	0.72	0.47	3.68
7	5.93	8.21	0.38	0.86	0.50	4.44
8	3.06	9.67	8.61	1.55	1.21	4.00
9	8.33	12.12	27.18	1.20	0.60	10.47
10	11.19	12.52	30.22	0.61	0.56	13.12
11	12.08	15.38	82.27	0.78	0.70	8.61
12	12.05	54.92	15.47	0.71	0.74	12.39
13	12.18	12.63	12.45	0.78	0.64	13.14
14	12.04	20.36	0.84	3.25	0.68	13.50
15	12.62	16.01	12.21	10.25	4.12	62843.85
16	13.12	11.22	13.08	12.90	1.57	13.63

Table 3: TBE-Raw performance per digit length (DL). The higher the values, the further the predicted values are from the ground truth values.

4 Experimental Results

4.1 RQ1: How does the y-axis information affect MLM performance in chart-to-table translation?

Biases by digit length (measured by RMS_{F1}^{TBE}): Figure 3 shows performance variation by digit length using our FairChart2Table dataset Part A. There are 60 tables per digit length. DePlot's performance generally declines as the digit length increases, except for the case of digit length 0. The performance at digit length 1 is the best, which is also the case for charts with the maximum y-axis value in the range [1, 10). ChartGemma exhibits relatively low overall performance among the compared models, with a general downward trend as digit length increases up to 10, followed by a slight recovery at larger digit lengths. TinyChart's performance generally decreases with increasing digit lengths. Extremely low performance occurs at digit lengths 11, 15, and 16. Pixtral exhibits a similar trend to TinyChart, with the decrease being largest at digit lengths of 15 and 16. Compared to the other models, Gemini is the least biased and achieves

Model	Base Conf.	#Major ticks		Range			Format		
		3	11	Pos	Neg	Ext	Comma	Sci.	Abbr.
Open Source									
DePlot	45.48	27.88	47.94	46.81	29.19	9.07	63.79	17.76	21.61
ChartGemma	19.34	6.68	21.45	9.25	9.92	3.43	20.77	1.38	9.92
TinyChart	72.56	33.24	67.45	25.17	7.00	45.86	37.11	80.65	26.29
Pixtral	67.88	29.16	79.11	42.43	11.15	31.63	58.92	71.89	32.14
Closed Source									
Gemini	83.01	41.77	85.58	80.22	71.68	44.32	83.26	82.62	46.54
GPT-4o	8.84	6.50	10.75	10.44	11.91	5.47	14.12	13.44	6.19
Prompting with major tick values (Ours)									
Pixtral	76.62	44.14	85.13	49.47	14.86	37.01	73.97	72.83	58.74
Gemini	83.77	42.12	85.61	79.71	74.42	49.91	83.80	82.90	61.32
GPT-4o	13.22	11.00	16.31	12.90	13.38	7.01	12.33	12.29	11.10

Table 4: RMS_{F1}^{TBE} performance for the base configuration (with only 3 entities, 6 major ticks, the x-axis and y-axis intersecting at the origin, and the plain numeric tick value format) and others. Statistically significant cases were highlighted. Cyan and pink indicate that the base configuration is better or worse than the compared configurations, respectively.

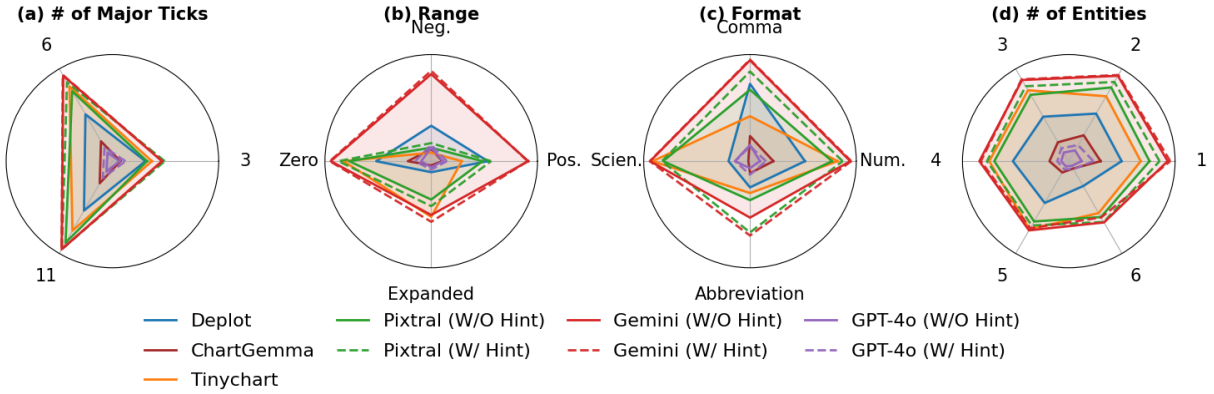

Figure 4: RMS_{F1}^{TBE} performance for bias evaluation associated with the number of major ticks, y-axis value range, tick number format, and the number of entities

the highest performance with digit lengths ranging from 6 to 16. However, its performance still decreases with increasing digit lengths. GPT-4o performs the worst among the compared models and still exhibits bias by digit lengths.

To what extent models are affected by digit length (measured by TBE-Raw): High values of TBE-Raw indicate large errors. Table 3 shows that at some digit lengths, there are unusually large errors. At digit length 15, GPT-4o has an extremely large error, differing from that at digit length 16, but the corresponding RMS_{F1}^{TBE} in Figure 3 are both under 15%. DePlot's raw error at digit length 0 is relatively larger than those at digit lengths 1-4. Similar patterns are observed with ChartGemma at digit length 12, TinyChart at digit lengths 9 and 10, and with Gemini at digit length 15.

Biases by the number of major ticks (measured by RMS_{F1}^{TBE}): Our FairChart2Table Part B was used to evaluate the effect of the number of major ticks of 3 and 11. Recall that the base configuration is from Part A, with 6 major ticks. Figure 4(a) and Table 4 demonstrate interesting patterns. All the models perform the worst with 3 major ticks. In contrast, for the 11 major ticks, the performance of the closed-source models improves compared to the base configuration, but there is no uniform trend for the open-source models.

Biases by the range of y-axis values (measured by RMS_{F1}^{TBE}): We used Part C of our dataset for the experiments. As shown in Figure 4(b) and Table 4, DePlot's performance slightly increases when the intersection of the y-axis with the x-axis shifts from zero to a positive value, decreases when it moves from zero to a negative value, and is lowest for charts with extended ranges. TinyChart performs the best when the y-axis crosses the x-axis at zero, outperforming DePlot under this base config-

# of Entities	DePlot			ChartGemma			TinyChart			Pixtral			Gemini			GPT-4o		
	F1 ↑	Sig ↑	SES ↓	F1 ↑	Sig ↑	SES ↓	F1 ↑	Sig ↑	SES ↓	F1 ↑	Sig ↑	SES ↓	F1 ↑	Sig ↑	SES ↓	F1 ↑	Sig ↑	SES ↓
1–2	47.78	50.00	-1.03	27.38	34.55	3.38	65.14	69.90	0.05	73.57	85.23	-0.44	87.58	97.27	1.06	11.25	21.34	3.84
3–4	47.59	50.27	-2.33	18.26	24.41	7.18	71.62	77.87	-0.14	67.03	81.26	-2.74	81.23	94.26	-0.62	7.78	15.07	7.13
5–6	34.10	36.96	3.10	9.57	14.32	11.42	61.37	68.56	4.06	59.38	75.57	-3.96	66.69	81.56	0.67	5.08	9.94	9.60

Table 5: Performance by entity count. All metric values are reported as percentages; F1 denotes RMS_{F1}^{TBE}; Sig. denotes RMS_{F1}^{TBE}-Sig.

uration. However, it performs significantly worse than DePlot on the chart, with a displaced y-axis origin. Additionally, TinyChart handles charts with extended ranges better than it does charts with a displaced y-axis origin. ChartGemma's performance drops substantially when the y-axis origin shifts from zero to a positive value, decreases slightly further when it shifts to a negative value, and is lowest for charts with extended ranges. Pixtral performance decreases from zero to positive and then to negative y-axis displacement. In contrast, Pixtral outperforms DePlot at the extended ranges. Gemini performs consistently well regardless of the y-axis origin's displacement. Its performance is slightly better with a positive displacement and slightly worse with a negative displacement compared to no displacement. In contrast, a huge drop is observed for charts with extended ranges. On the contrary, GPT-4o performs better for charts with a displaced y-axis origin, compared to charts with the y-axis at the origin, while it performs worse when the range is extended.

Biases by the number format on the y-axis: The dataset Part D was used for evaluation. Figure 4(c) shows the results. Interestingly, DePlot performs better with the comma format than with the plain numerical format, while its performance significantly declines with the scientific notation and abbreviation formats. ChartGemma performs slightly better with the comma format than with the numerical format, but its performance drops sharply with the abbreviation format and reaches its lowest level with the scientific notation format. TinyChart performs slightly better with the scientific notation format than with the plain numerical format, while its performance significantly declines with the comma and abbreviation formats. Pixtral performs slightly better with the scientific format than with the plain numerical format. While its performance also declines with the comma and abbreviation formats, the drop is moderate for the comma format but dramatic with the abbreviation format. Gemini performs slightly better with the scientific notation format than with the plain numerical format, while its performance slightly decreases with the comma format. In contrast, there is a dramatic performance drop with the abbreviation format. Table 4 shows that GPT-4o performs better with the comma and scientific notation formats than with the plain numerical format, while its performance slightly decreases with the abbreviation format.

Entities	Average Number of Crossing Points
2	1.450
3	3.067
4	4.175
5	6.120
6	7.467

Table 6: Average number of crossing points per entity, grouped by number of entities from Part A (different digit length) of FairChart2Table.

4.2 RQ2: What other factors of chart images impact MLM performance?

Biases by the number of entities: Figure 4(d) shows a general trend that the model performance generally decreases as the number of entities increases, especially DePlot, ChartGemma, and Pixtral. In contrast, Gemini and TinyChart are more robust, although TinyChart shows a bigger performance drop at 6 entities compared to Gemini. As the number of entities increases, models are more likely to confuse numerical values on the same x-axis because of the higher chance of crossing points. We did not explicitly control the number of line crossings when varying the number of entities. However, Table 6 shows the high correlation between the number of entities and the average number of crossing points per entity with the Pearson correlation of 0.9549. This strong correlation suggests that, in our setup, line crossings are effectively mediated by the entity count, allowing SES to be a meaningful metric for this evaluation. Low SES scores are desirable. As shown in Table 5, DePlot has a high SES score with 5 and 6 entities, TinyChart and GPT-4o also show a dramatic increase with 6 entities, and ChartGemma shows a gradual increase as the number of entities increases.

In contrast, Pixtral's SES score decreases as the number of entities increases, and Gemini shows only a modest increase.

Type	DePlot	Chart Gemma	Tiny Chart	Pixtral	Gemini	GPT-4o
Line	41.64	12.97	65.12	55.81	72.01	8.39
Dot	46.13	12.70	67.91	72.30	81.87	6.93
Bar	41.69	29.53	65.09	71.85	81.62	8.76

Table 7: RMS_{F1}^{TBE} across different chart types.

RMS_{F1}^{TBE}-Sig by Entity Count: As shown in Table 5, the gap between RMS_{F1}^{TBE}-Sig and RMS_{F1}^{TBE} increases for all models except for ChartGemma and GPT-4o, indicating that the increasing prediction errors are still within the minor tick threshold. In contrast, for ChartGemma and GPT-4o, the gap decreases, implying more prediction errors fall outside the threshold. Both metric values fall, and the gap narrows.

Impact of chart types: MLM performance appears to vary across chart types, as reported by Kim et al. (2025). However, as shown in Table 7, when the underlying data for each chart type are the same, the model performance differences become less pronounced but still statistically significant except for three cases shown in Appendix A.3.

4.3 RQ3: Does prompting MLMs with explicit y-axis information help improve MLM performance?

For general-purpose MLMs, we introduce a y-axis hinting method that enables the model to perform chart-to-table translation conditioned on manually extracted y-axis tick values. See the prompts in Appendix A.4. Table 4 illustrates that Pixtral exhibits substantial gains using our y-axis tick-value hints, ranging from 0.94 to 26.60. Moreover, the prompting strategy also enhances GPT-4o's performance, except for the comma and scientific notation formats. In contrast, Gemini performs generally similarly or only slightly better with the additional information, but when labels are abbreviated, its performance increases dramatically, with the score rising from 46.54 to 61.32.

5 Conclusions and Future Work

We propose the FairChart2Table benchmark and demonstrate biases related to the y-axis information for chart-to-table translation. Our benchmark can be used to measure similar biases in other MLMs. Using manually extracted y-axis tick values as prompt hints can significantly reduce the bias related to digit lengths. Future work includes automated debiasing methods for chart-to-table translation tasks.

6 Limitations

This paper provides a comprehensive evaluation of how y-axis information affects MLM performance in chart-to-table translation. It has the following limitations. First, the proposed dataset consists of three chart types and does not include real-world chart images. However, to study bias associated with one aspect, such as digit length, we need to keep the other aspects fixed and balanced to prevent the confounding impact from the other aspects. Synthetic data generation is required. Second, charts and legends are only in English, as in existing research (Liu et al., 2023a). Lastly, this study does not focus on biases unrelated to the y-axis chart-to-table translation. Our FairChart2Table dataset and framework can be extended to include other chart types and non-English headers.

7 Ethical Considerations and Potential Risk

The authors follow the ACL Code of Ethics. The ideas presented in this paper regarding investigating y-axis-related biases, experimental designs, and conclusions are solely those of the authors. AI assistants were utilized only for writing refinement, presentation, and basic coding support, including formatting and debugging. Since the dataset is synthetically generated, it is not subject to privacy concerns. All scripts, evaluation metrics, datasets, and prompting methods will be made publicly available for reproducibility. Our findings show that the degree of bias varies across different understudied MLMs. Therefore, specific findings may not generalize to other MLMs or to the same models under different training or fine-tuning conditions. Nevertheless, our FairChart2Table framework can be used to investigate the y-axis biases in other MLMs in a similar way.

8 Acknowledgement

This work is partially supported by the National Science Foundation under Grant No. 2152117. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, et al. 2024. Pixtral 12b. *arXiv preprint arXiv:2410.07073*.

Mubashara Akhtar, Oana Cocarascu, and Elena Simperl. 2023. Reading and reasoning over chart images for evidence-based automated fact-checking. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 399–414, Dubrovnik, Croatia. Association for Computational Linguistics.

Ritwick Chaudhry, Sumit Shekhar, Utkarsh Gupta, Pranav Maneriker, Prann Bansal, and Ajay Joshi. 2020. Leaf-qa: Locate, encode attend for figure question answering. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3501–3510.

Ashim Gupta, Vivek Gupta, Shuo Zhang, Yujie He, Ning Zhang, and Shalin Shah. 2024. Enhancing question answering on charts through effective pre-training tasks. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 185–192, Miami, Florida, US. Association for Computational Linguistics.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Mohammed Saidul Islam, Raian Rahman, Ahmed Masry, Md Tahmid Rahman Laskar, Mir Tafseer Nayeem, and Enamul Hoque. 2024. Are large vision language models up to the challenge of chart comprehension and reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3334–3368, Miami, Florida, USA. Association for Computational Linguistics.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5648–5656.

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

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

Wonjoong Kim, Sangwu Park, Yeonjun In, Seokwon Han, and Chanyoung Park. 2025. Simplot: Enhancing chart question answering by distilling essentials. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 573–593.

Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2023a. Deplot: One-shot visual language reasoning by plot-to-table translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10381–10399.

Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Eisenschlos. 2023b. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12756–12770.

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

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279.

Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. 2023. Unichart: A universal vision-language pretrained model for chart comprehension and reasoning. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 14662–14684.

Ahmed Masry, Mehrad Shahmohammadi, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. 2024. ChartInstruct: Instruction tuning for chart comprehension and reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10387–10409, Bangkok, Thailand. Association for Computational Linguistics.

Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. 2025. ChartGemma: Visual instruction-tuning for chart reasoning in the wild. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 625–643, Abu Dhabi, UAE. Association for Computational Linguistics.

Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2024. Chartassistant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7775–7803.

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.

Srija Mukhopadhyay, Adnan Qidwai, Aparna Garimella, Pritika Ramu, Vivek Gupta, and Dan Roth. 2024. Unraveling the truth: Do VLMs really understand charts? a deep dive into consistency and robustness. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16696–16717, Miami, Florida, USA. Association for Computational Linguistics.

Karl Pearson. 1896. Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, (187):253–318.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2024. Spiqa: A dataset for multimodal question answering on scientific papers. *Advances in Neural Information Processing Systems*, 37:118807–118833.

Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. 2024. TinyChart: Efficient chart understanding with program-of-thoughts learning and visual token merging. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1898, Miami, Florida, USA. Association for Computational Linguistics.

Zifeng Zhu, Mengzhao Jia, Zhihan Zhang, Lang Li, and Meng Jiang. 2025. Multichartqa: Benchmarking vision-language models on multi-chart problems. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11341–11359.

A Appendix

A.1 Details about RMS_{F1}^{TBE}

The similarity score with the distance between two numerical values is defined as:

$$D_{\text{TBE}}(g_j, p_i) = min\left(1, \frac{|g_j - p_i|}{t}\right), \quad (6)$$

where $\mathbf{g} = \{g_j\}_{1 \le j \le m}$ for the ground truth column with m elements and $\mathbf{p} = \{p_i\}_{1 \le i \le n}$ for the predicted column with n elements, respectively. In our implementation, the value of t is one-fifth of the interval between consecutive major ticks of a given chart image. Following the minimal cost matching procedure used in RMS (Liu et al., 2023a), we align the predicted and ground-truth datapoints using only *Normalized Levenshtein Distance* between their row and column headers. Let $X \in \{0,1\}^{n \times m}$ denote the resulting binary assignment matrix, where $X_{ij} = 1$ if the predicted header of datapoint i is matched to the ground-truth header of datapoint j, and $X_{ij} = 0$ otherwise. Once the assignment is fixed, we compute precision and recall using only $D_{\text{TBE}}(g_j, p_i)$ for the matched value pairs, without incorporating header similarity into the final value-level score.

$$S_{\text{TBE}}(g_j, p_i) = 1 - D_{\text{TBE}}(g_j, p_i) \quad (7)$$

$$\text{RMS}_{precision}^{TBE} = \frac{\Sigma_i^n \Sigma_j^m X_{ij} S_{\text{TBE}}(g_j, p_i)}{n} \quad (8)$$

$$\text{RMS}_{recall}^{TBE} = \frac{\Sigma_i^n \Sigma_j^m X_{ij} S_{\text{TBE}}(g_j, p_i)}{m} \quad (9)$$

RMS_{F1}^{TBE} is the harmonic mean of the above precision and recall.

A.2 Results of RMS_{F1} with and without header similarity scores

As Figures 5 illustrates, some models are over-penalized a lot because of headers. We compare RMS_{F1} and RMS_{F1} without header score on FairChart2Table data. As the figure illustrates, different models generate distinct outputs for various headers, and that affects the accuracy.

A.3 Result of Statistical Test for Impact of Chart Types on MLM performance

See Table 8.

A.4 Prompts for Gemini and Pixtral

See Tables 9 - 10.

A.5 Examples FairChart2Table Chart Images

A.5.1 Part A

See Figures 6 - 9.

A.5.2 Part B

See Figures 10 - 18.

A.5.3 Part C

See Figures 12 - 14.

A.5.4 Part D

See Figures 15 - 17.

A.6 Examples of Original Chart and Generated Chart Based on Incorrectly Predicted Tables By DePlot

See Figures 19 - 22.

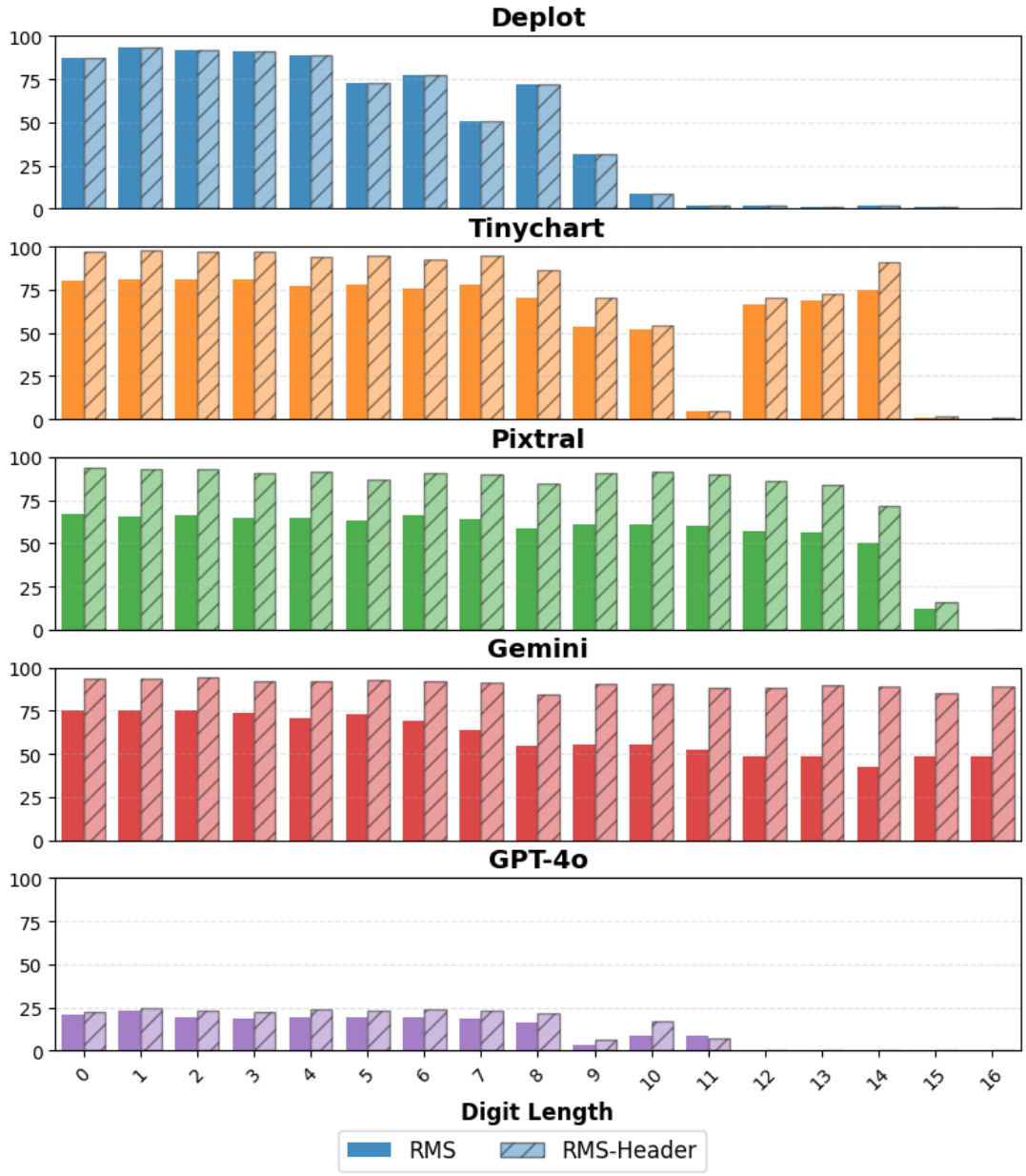

Figure 5: Bar charts comparing RMS_{F1} and RMS-Header (RMS_{F1} without header scores) by different models.

Type	DePlot	ChartGemma	Tiny	Pixtral	Gemini	GPT
Dot VS. Bar	5.067968e-02	1.85467e-67	4.041066e-02	1.318864e-02	2.032969e-08	5.515060e-18
Line VS. Bar	7.570325e-14	5.44866e-72	2.044370e-28	1.147831e-108	8.373813e-126	1.269979e-01
Line VS. Dot	1.419833e-07	0.871489	5.968554e-19	4.882375e-110	6.484573e-118	2.871905e-12

Table 8: P-values of the Wilcoxon Signed-rank test. Except for Dot vs. Bar with Deplot and Line vs. Bar with GPT, all p-values are smaller than 0.05. Yellow indicates that the comparison is statistically significant.

Prompt:
Generate underlying data table for the chart.

Table 9: Prompt for chart-to-table translation

Prompt:
Generate underlying data table for the chart. Hint: y-axis major ticks are ...

Table 10: Prompt for our hint strategy using major tick values in scientific notation for chart-to-table translation

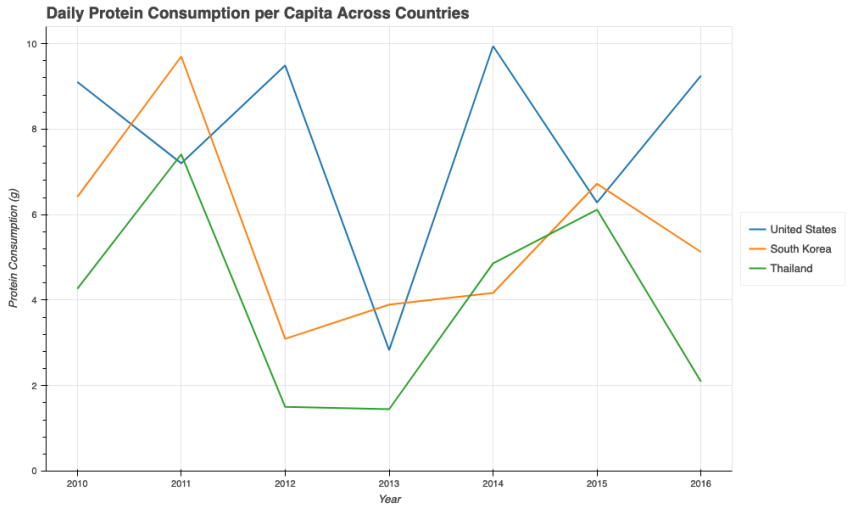

Figure 6: Part A: Line Chart

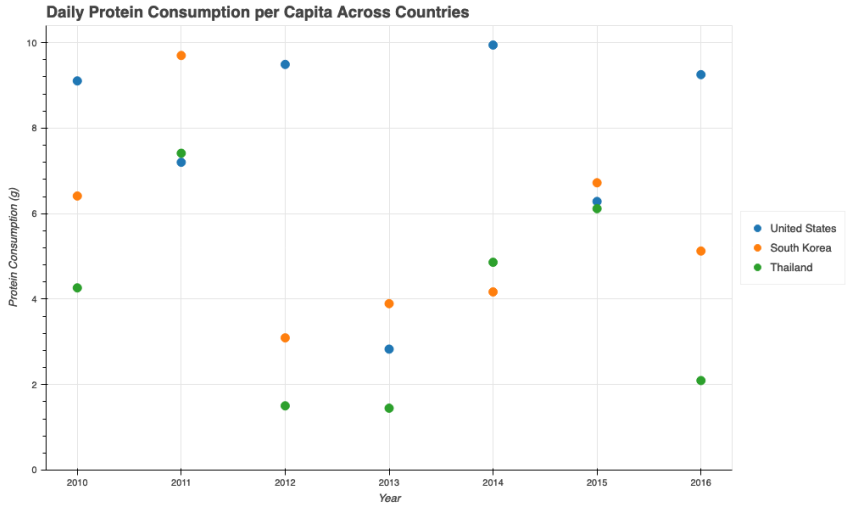

Figure 7: Part A: Dot Chart

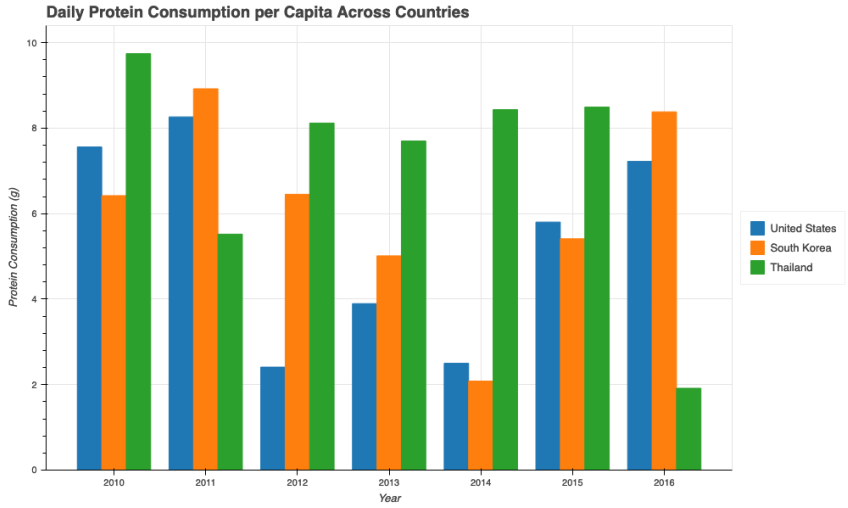

Figure 8: Part A: Bar Chart

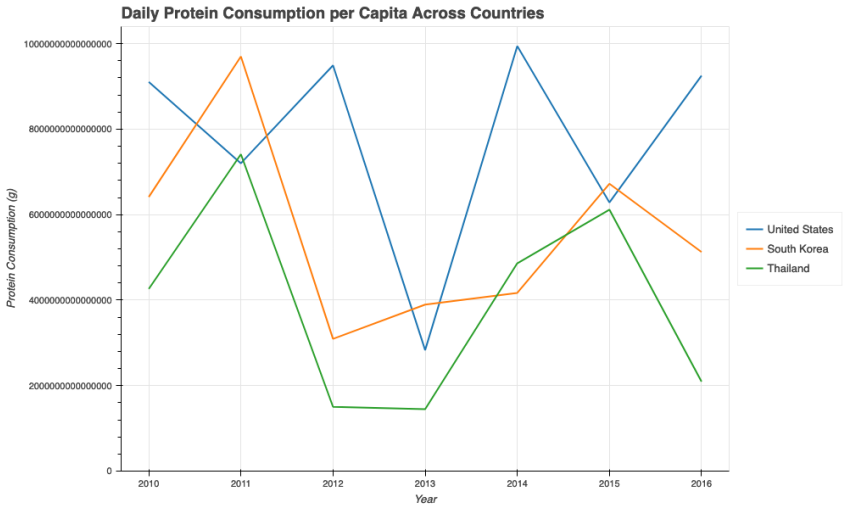

Figure 9: Part A: Line Chart at Digit Length 16

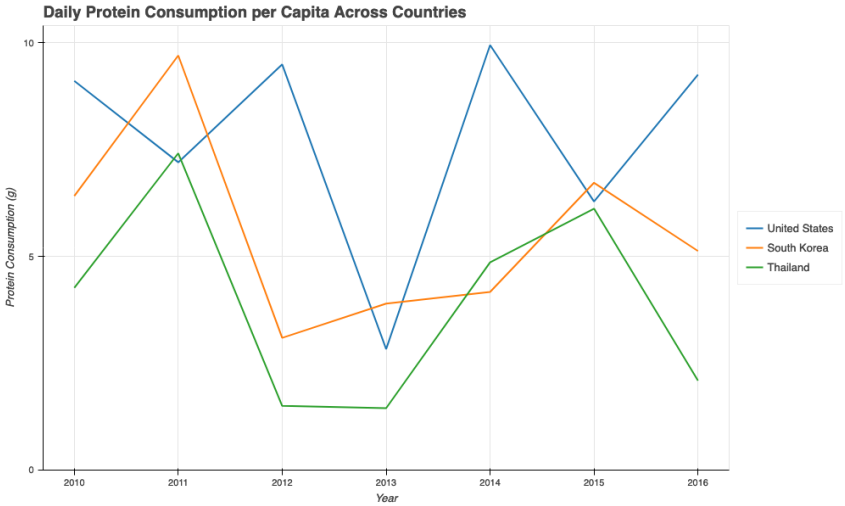

Figure 10: Part B: Line Chart at Digit Length 1 with 3 Major Ticks

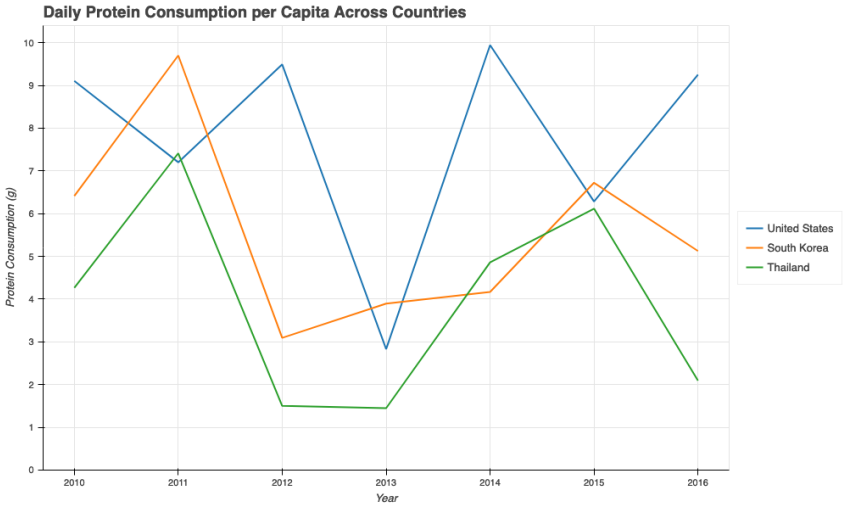

Figure 11: Part B: Line Chart at Digit Length 1 with 11 Major Ticks

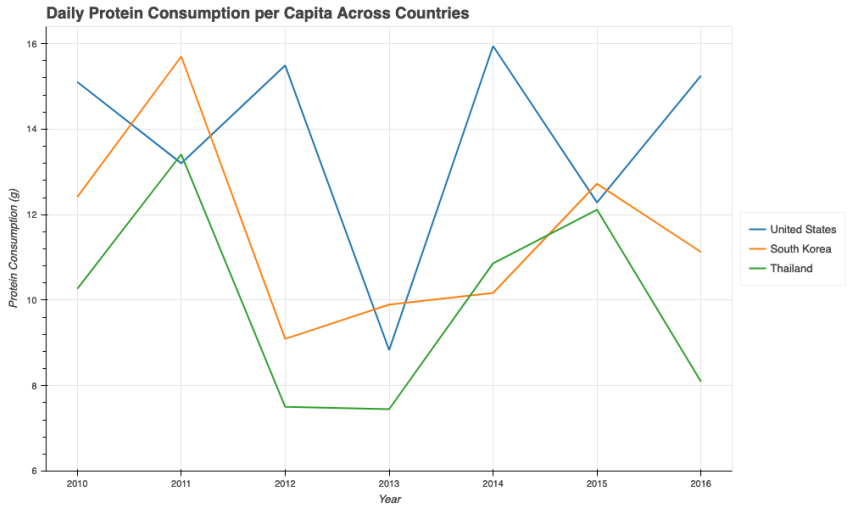

Figure 12: Part C: Line Chart with Positive Minimum Tick Value transformed from Digit Length 1

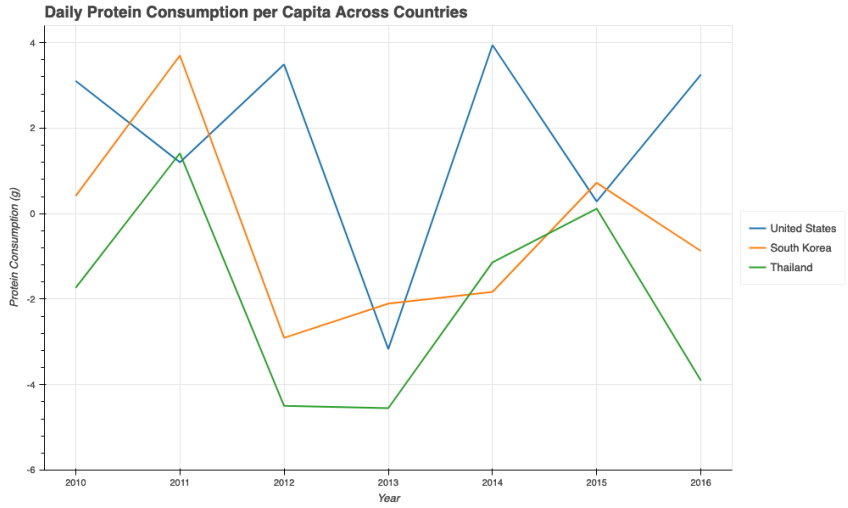

Figure 13: Part C: Line Chart with Negative Minimum Tick Value transformed from Digit Length 1

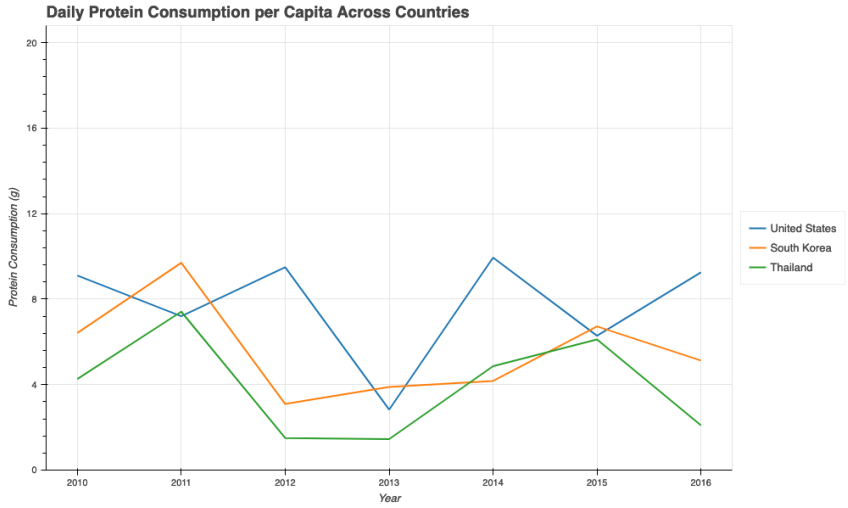

Figure 14: Part C: Line Chart at Digit Length 1 with Extended Range

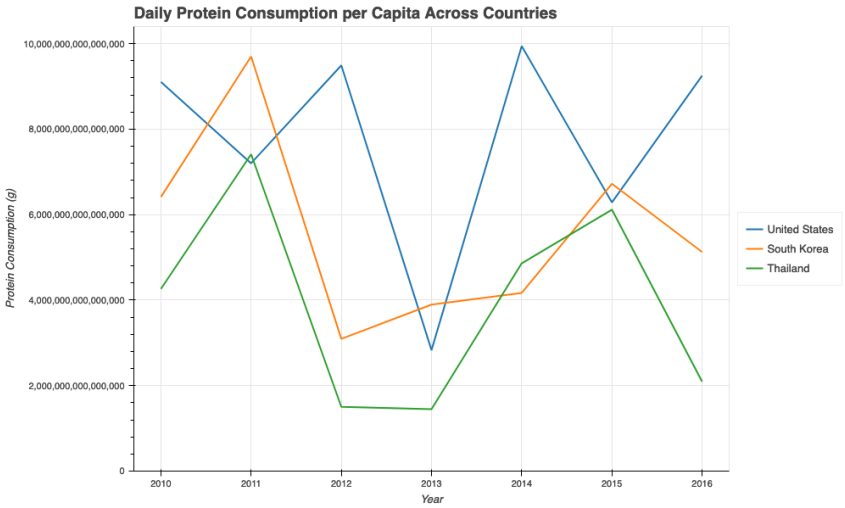

Figure 15: Part D: Line Chart at Digit Length 16 with Comma Format

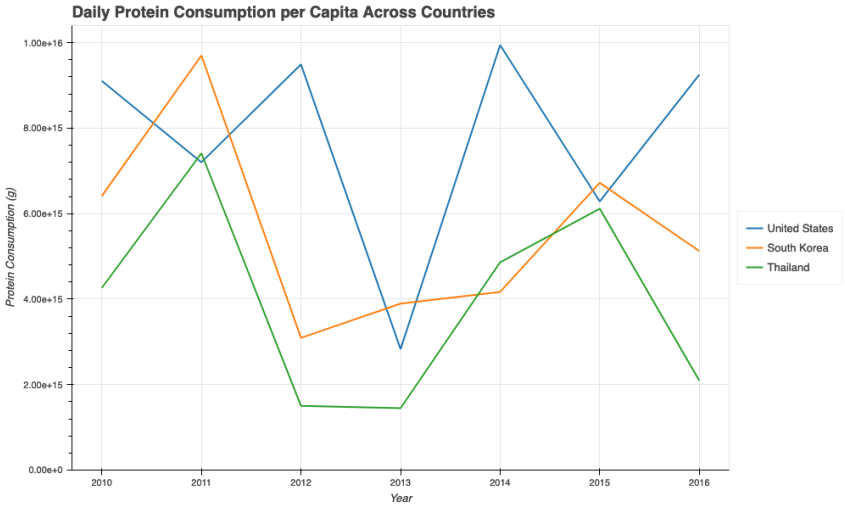

Figure 16: Part D: Line Chart at Digit Length 16 with Scientific Notation Format

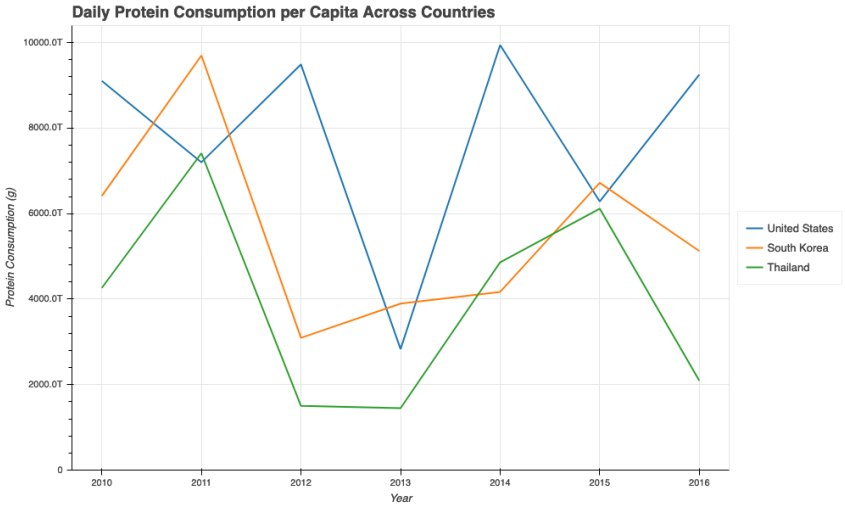

Figure 17: Part D: Line Chart at Digit Length 16 with Abbreviation Format

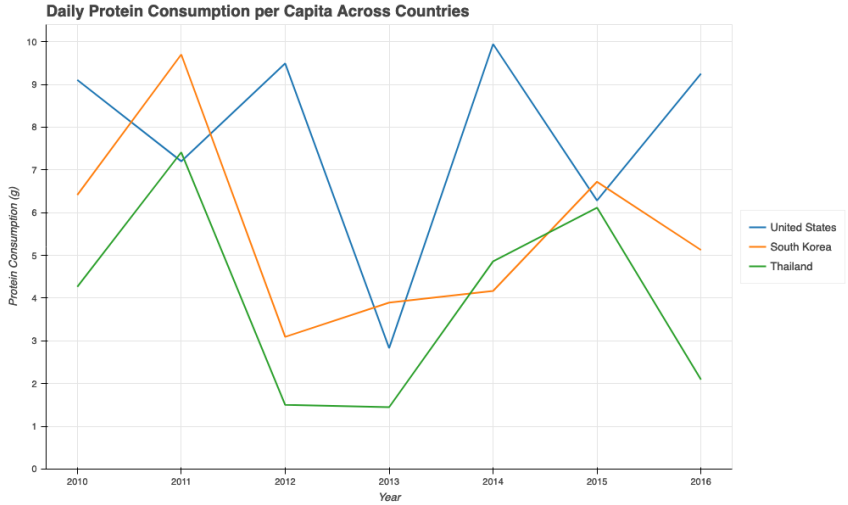

Figure 18: Part B: Line Chart at Digit Length 1 with 11 Major Ticks

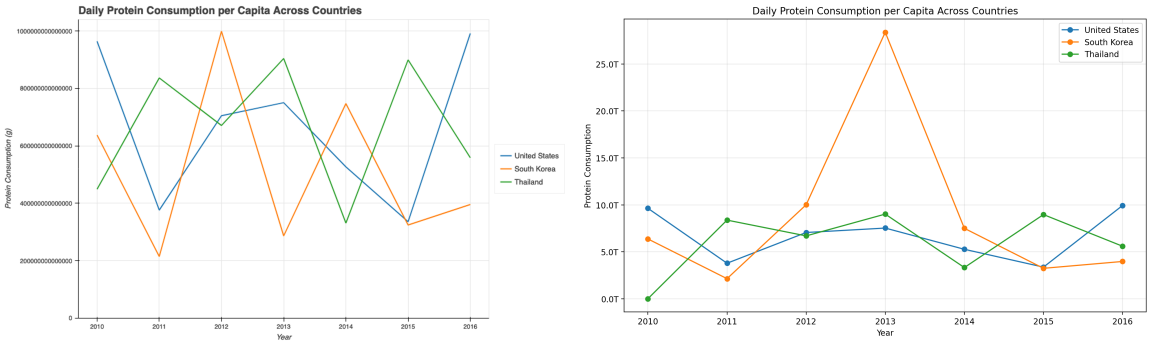

Figure 19: Comparison between the original chart (left) and the chart predicted by DePlot (right). The predicted chart reflects a long-digit-length bias, returning values with incorrect digit lengths.

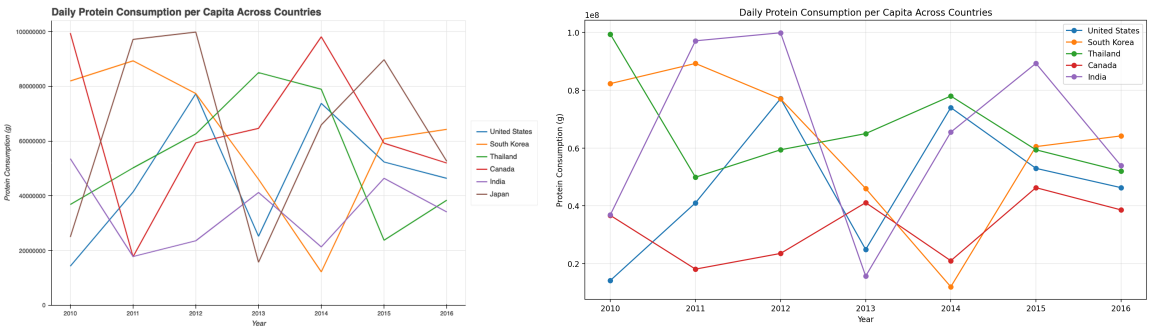

Figure 20: Comparison between the original chart (left) and the chart predicted by DePlot (right). The predicted chart shows confusion caused by many entities and frequent line crossings, leading to mismatched values across categories.

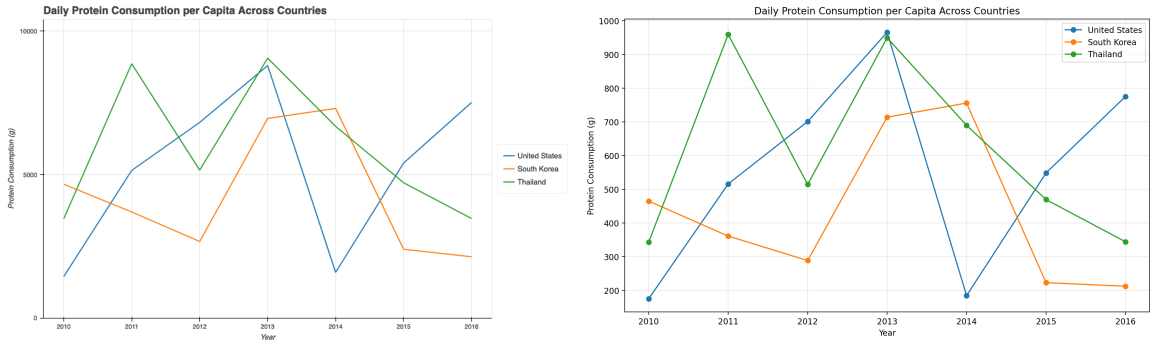

Figure 21: Comparison between the original chart (left) and the chart predicted by DePlot (right). With only three major ticks, the predicted chart preserves a similar overall shape but distorts the proportions.

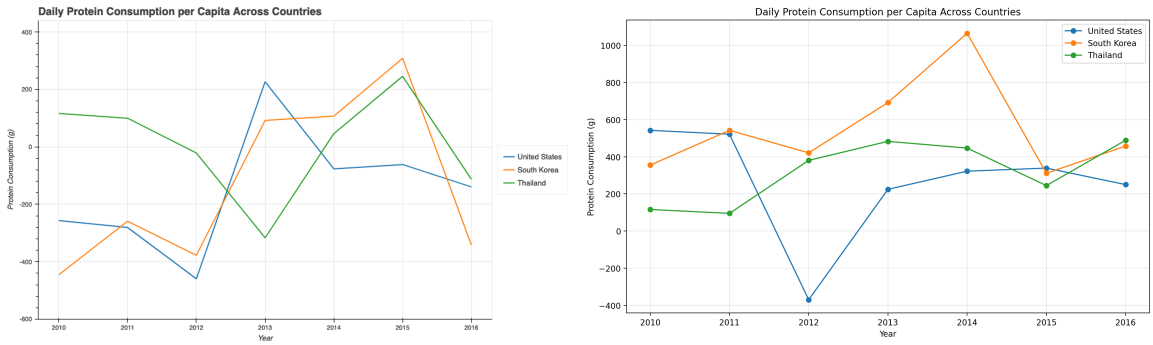

Figure 22: Comparison between the original chart (left) and the chart predicted by DePlot (right). When the y-axis origin is negative, the predicted chart misestimates the point values.