ChartCards: A Chart-Metadata Generation Framework for Multi-Task Chart Understanding

Yifan Wu*
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
ywu012@connect.hkust-gz.edu.cn

Lutao Yan*
South China University of Technology
Guangzhou, China
lutaoyan666@gmail.com

Leixian Shen
The Hong Kong University of Science and Technology
Hong Kong SAR, China
lshenaj@connect.ust.hk

Yinan Mei
Huawei Cloud BU
Hangzhou, China
yinan.mei@huawei.com

Jiannan Wang
Huawei Cloud BU, Simon Fraser University
Hangzhou, China
jnwang@sfu.ca

Yuyu Luo
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
yuyuluo@hkust-gz.edu.cn

Abstract

The emergence of Multi-modal Large Language Models (MLLMs) presents new opportunities for chart understanding. However, due to the fine-grained nature of these tasks, applying MLLMs typically requires large, high-quality datasets for task-specific fine-tuning, leading to high data collection and training costs. To address this, we propose ChartCards, a unified chart-metadata generation framework for multi-task chart understanding. ChartCards systematically synthesizes various chart information, including data tables, visualization code, visual elements, and multi-dimensional semantic captions. By structuring this information into organized metadata, ChartCards enables a single chart to support multiple downstream tasks, such as text-to-chart retrieval, chart summarization, chart-to-table conversion, chart description, and chart question answering. Using ChartCards, we further construct MetaChart, a large-scale high-quality dataset containing 10,862 data tables, 85K charts, and 170K high-quality chart captions. We validate the dataset through qualitative crowdsourcing evaluations and quantitative fine-tuning experiments across various chart understanding tasks. Fine-tuning six different models on MetaChart resulted in an average performance improvement of 5% across all tasks. The most notable improvements are seen in text-to-chart retrieval and chart-to-table tasks, with Long-CLIP and Llama 3.2-11B achieving improvements of **17%** and **28%**, respectively.

ACM Reference Format:
Yifan Wu, Lutao Yan, Leixian Shen, Yinan Mei, Jiannan Wang, and Yuyu Luo. 2025. ChartCards: A Chart-Metadata Generation Framework for Multi-Task Chart Understanding. In . ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Both authors contributed equally to this research.

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Charts, representing complex data visually, are widely used in daily life [6, 38]. However, as the volume of charts increases and their formats become more diverse, traditional manual analysis methods struggle to meet the growing demands for both efficiency and accuracy. To address this, automated chart understanding has emerged as a key solution [32, 39, 45, 46]. In recent years, the rapid development of Multimodal Large Language Models (MLLMs), such as OpenAI-o1 [37], Flamingo [1], BLIP2 [18] and CLIP [8], has provided new methods for automated chart understanding. By jointly learning visual and linguistic information, these models have demonstrated powerful cross-modal understanding capabilities [46, 51].

Despite their success in general tasks, adapting MLLMs for chart understanding remains challenging [23, 40, 55]. Chart understanding tasks need finer-grained processing capabilities, often requiring additional large-scale, high-quality chart-specific data to train or fine-tune models. Moreover, chart understanding involves multiple downstream tasks, such as text-to-chart retrieval [47], chart description [49], chart question answering (ChartQA) [12, 14, 34, 50], chart-to-table [23], and chart summary [15, 41]. Each task imposes distinct data annotation requirements. For example, ChartQA tasks require precise numerical annotations, while summarization tasks focus more on semantic information descriptions. This annotation heterogeneity complicates data organization and increases the cost of cross-task dataset integration.

Recently, researchers have introduced various chart-related datasets to advance chart understanding, as summarized in Table 1. However, problems remain in three key aspects: metadata and task diversity, dataset size and scalability, and quality validation. Regarding metadata and task diversity, many datasets provide limited metadata, restricting them to a single chart understanding task [12, 14, 15, 34, 36, 41] or a narrow set of multi-task applications [35]. In terms of dataset size and scalability, while some multi-task datasets support diverse chart understanding tasks, their overall size remains limited [9, 49], or they are constructed from existing data datasets, limiting their scalability [33, 35]. For quality validation, many studies rely on closed-set evaluation rather than cross-dataset evaluation [11, 49], potentially introducing bias. Additionally, most datasets lack human-based evaluation [9, 11, 33, 35, 49].

Table 1: Comparison with Existing Chart-related Training Datasets (DT means Data Tables, VE means Visual Elements, VC means Visualization Code, and CC means Chart Captions).

Task Diversity	Datasets	Metadata					#-Chart	Data	Quality Evaluation	
		#DT	#VE #	#VC	#CC	#Charts	Types	Generation	Benchmark	Human Check
Single Task	DVQA [12]	200K	/	/	/	200K	1	Rule-Based	Closed-Set	✗
	ChartQA [34]	20.8K	20.8K	/	/	20.8K	3	Human-Based	Closed-Set	✓
	PlotQA [36]	224K	224K	/	/	224K	3	Rule-Based	Closed-Set	✗
	Chart-to-text [15]	44k	/	/	44k	44k	6	LLM-Based	Closed-Set	✗
	Vistext [41]	12k	/	/	24k	12K	3	Human-Based	Closed-Set	✓
	FigureQA [14]	100K	/	/	/	100K	5	Rule-Based	Closed-Set	✗
Multi Tasks	Unichart [33]	601K	/	/	481K	611K	3	Collection-Based	Cross-Dataset	✗
	ChartLlama [9]	11K	/	/	/	11K	10	LLM-Based	Cross-Dataset	✗
	ChartSFT [35]	911K	/	/	1M	39M	9	Collection-Based	Cross-Dataset	✗
	NovaChart [11]	47K	47K	47K	/	47K	18	LLM-Based	Closed-Set	✗
	MetaChart (ours)	85K	85K	85K	170K	85K	11	LLM-Based	Cross-Dataset	✓

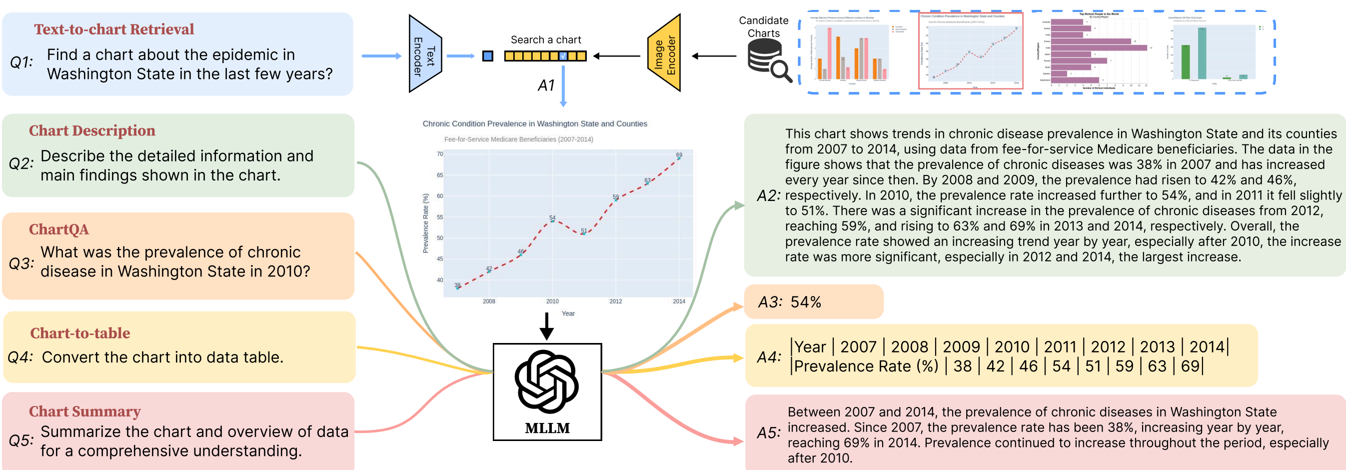

Figure 1: Example of five downstream tasks corresponding to a chart: Text-to-chart retrieval (Q1), chart description (Q2), ChartQA (Q3), chart-to-table (Q4), and chart summary (Q5). Each task will produce the corresponding result, A1 to A5.

To address the issues, we propose ChartCards, a novel chart-metadata generation framework for multi-task chart understanding. First, we design a unified structured representation format for chart metadata, including data table, visualization code, visual element information, analytic task, and captions for chart overview and chart analysis. This format enables direct extraction from a single chart and supports multiple downstream tasks, such as text-to-chart retrieval, chart-to-description, chart question answering, chart-to-table, and chart-to-summary, as shown in Figure 1. Next, we introduce a parameterized programmable chart generation pipeline, which facilitates the rapid creation of large-scale, diverse chart metadata through flexible configuration options. It overcomes the speed limitations of manual data collection and significantly enhances scalability. Leveraging the ChartCards framework, we establish a large-scale high-quality dataset, MetaChart, consisting of 10,862 data tables, 85K raw charts, and 170K high-quality chart captions. Finally, we implement a dual-quality verification mechanism for the generated dataset. On the one hand, we conduct crowdsourced evaluations to assess generated chart captions' semantic and visual consistency. On the other hand, we validate the framework's effectiveness through extensive fine-tuning cross-dataset experiments, demonstrating substantial performance improvements across multiple downstream chart understanding tasks.

Contributions. The main contributions of this paper are:

(1) **Chart-Metadata Generation Framework.** We propose ChartCards, an automated and scalable chart-metadata generation framework, capable of efficiently generating large-scale data for multi-task chart understanding.

(2) **MetaChart Dataset.** Using the ChartCards framework, we build a large-scale high-quality dataset, MetaChart, including 10,862 data tables, 85K charts, and 170K high-quality chart captions.

(3) **Extensive Experiments.** Through qualitative human-based evaluation and quantitative fine-tuning experiments on downstream tasks, we validate the effectiveness of our data generation framework and demonstrate that our MetaChart significantly enhances model performance. The six state-of-the-art models achieved an average improvement of 5% across five tasks.

2 Related Work

2.1 Chart Understanding Tasks and Datasets

In recent years, with the rapid development of MLLMs [4, 17, 19, 20, 27], cross-modal understanding technology based on charts has gradually become a research hotspot [42]. This field mainly revolves around the bidirectional semantic mapping between charts and texts, forming several challenging sub-task systems. Text-to-chart retrieval requires the model to find the most relevant chart for a query through semantic matching. CRBench [47], the first specialized benchmark, supports this research. Chart-to-table conversion extracts numerical data from visual elements, requiring strong spatial perception and structured data reconstruction. Datasets like StructChart [48] provide paired samples with precise annotations to train robust models. In the dimension of semantic abstraction, existing research divides it into two progressive levels: chart summary extracts key trends, while chart description provides detailed analysis. Besides the above four chart tasks, ChartQA [34] is a more comprehensive evaluation task requiring the model to integrate multi-dimensional capabilities such as visual parsing, semantic reasoning, and domain knowledge. However, existing chart datasets often support only 1-2 tasks. While NavoChart [49] enables multiple tasks, it lacks human evaluation and public dataset validation. ChartCards, which we proposed, is a scalable framework. The training data generated by this framework, MetaChart, has not only undergone quality validation through crowdsourcing experiments but also significantly improved the model's performance on downstream chart tasks.

2.2 Multi-modal Large Language Models

In recent years, MLLMs have made breakthrough progress in the field of cross-modal semantic alignment through pre-training architecture innovations like cross-modal attention mechanisms [3, 16, 52, 52] and post-training optimization strategies like [5, 10]. The research paradigm has gradually evolved from simple feature-level fusion to cognitive-level interaction. Typical works such as Flamingo [1] implement multimodal sequence modeling through gated cross-attention, while BLIP-2 [18] explores the application of lightweight adapters such as Q-Former in cross-modal alignment. These technical breakthroughs have significantly improved the models' performance on many natural image tasks [8, 21, 44]. In view of the great achievements of multimodal large models, many researchers use chart-related data to train and fine-tune these models so that multimodal models can obtain multiple chart understanding capabilities [24, 28, 49]. Unichart [33] crawled a large amount of data while integrating current public data sets in order to enable the model to complete multiple chart tasks such as chart-to-table [23]; ChartAssistant [35] pre-trained and fine-tuned Unichart as the base model, collected training data from 8 public data sets, and further expanded the training data through data enhancement, and finally successfully achieved SOTA on multiple tasks; Furthermore, TinyChart [54] collected data on nearly 20 public data sets, and the most popular model with 3B parameters exceeded the performance of ChartAssistant with 13B parameters on multiple benchmarks. Although performance can be improved by continuously increasing training data, it is very costly to collect chart data from multiple data sources. ChartCards proposed in this paper can generate training data in batches while reducing the cost of data production.

3 ChartCards

In this section, we will introduce the ChartCards framework in detail. First, we will explore the process of data collection and how to automatically generate diverse charts based on this data. Next, we will introduce the Chart Metadata generated by ChartCards and its specific information. Finally, we will introduce MetaChart, a large-scale and high-quality dataset built based on ChartCards.

3.1 Data Collection and Visualization

Step 1: Data Collection. The first step in generating high-quality semantic insights is selecting a reliable and diverse data source. To ensure the authenticity and usability of the data, we chose Kaggle [13] as our primary dataset provider. Most Kaggle datasets originate from real-world applications or competitions, and its rating system allows for effective initial filtering, ensuring that the selected datasets are well-structured, rich in content, and suitable for visualization.

As shown in Figure 2:❶. During the data selection phase, we retained only datasets with a rating higher than 8.0 to enhance data quality. To further improve data usability, we performed a cleaning and preprocessing step, which included: 1. Removing rows with missing values to ensure data integrity. 2. Discarding CSV files containing only textual information without numerical values to ensure usability for numerical analysis and visualization. 3. Eliminating duplicate columns with identical data reduces redundancy and enhances data quality.

After filtering, we collected a total of 10,862 CSV tables, covering a wide range of topics and statistical distributions.

Step 2: Splitting Tables Automatically. As shown in Figure 2:❷, we transform 10,862 CSV tables into 85*K* JSON files. We employed DeepEye [29–31], an advanced automatic visualization recommendation system, during the transformation. This system intelligently parses structured data and recommends the most suitable visualization types (*e.g.*, line charts, bar charts, scatter plots) based on data types and statistical features. Through this approach, DeepEye decomposes each CSV file into multiple corresponding JSON files, with each JSON file representing a complete visualization description, including data, visual encoding, and metadata.

Specifically, DeepEye first parses the data in the CSV files, identifies their structures and underlying patterns, and then generates multiple visualization schemes based on the data characteristics. Each scheme corresponds to an independent JSON file, which not only contains the raw chart data but also records the chart type, visual encoding configurations like legend data, and related metadata in detail. In this way, a single CSV file can generate multiple JSON files, each corresponding to a unique visual representation, significantly enhancing the diversity and coverage of the dataset.

Ultimately, we generated 85*K* JSON files, which provide a rich data foundation for subsequent chart understanding tasks (*e.g.*, chart summarization, chart captioning, chart question answering, and chart retrieval) and ensure the data's scalability and usability through a standardized format. Through this process, we efficiently

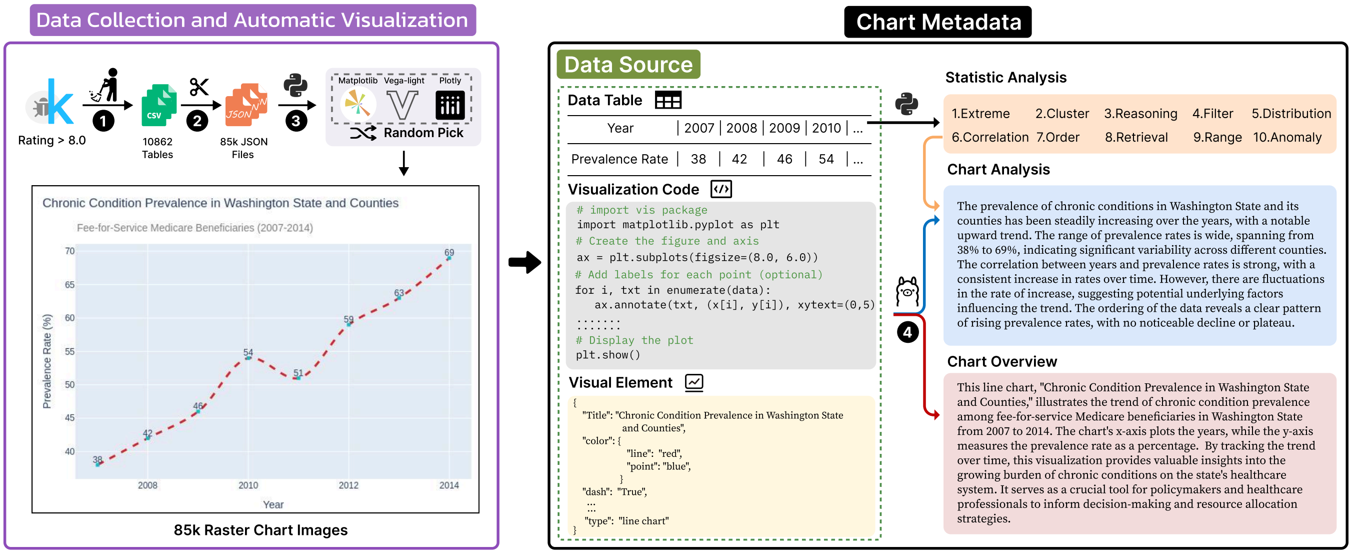

Figure 2: The ChartCards framework, a running example of the chart and corresponding metadata.

converted from structured data to multi-task chart understanding data, providing a high-quality and diverse dataset.

Step 3: Visualization Mapping Based on Templates. As shown in Figure 2:❸, the third step involves mapping the generated JSON files into visual charts. To further enhance the diversity of visual charts, we predefined three visualization syntax templates: Matplotlib, Vega-Lite, and Plotly, based on the JSON file format generated by DeepEye. These templates enable direct mapping of fields from JSON files to visualization code, allowing for the rapid generation of corresponding charts.

While designing these templates, we observed that many chart types, despite their significant visual differences, share minimal differences at the code level. For example, an area chart and a line chart differ by only a single boolean parameter. Leveraging this insight, we incorporated numerous similar configuration options into the three visualization syntax templates, enabling the generation of a wide variety of visually distinct chart types by adjusting a small set of parameters. This design not only significantly enhances the richness of visual charts but also simplifies the chart generation process, making the conversion from JSON files to visual charts more efficient and flexible.

Through this approach, we are able to generate multiple visual representations from the same JSON data, further expanding the diversity and applicability of the dataset and providing broader data support for subsequent chart understanding tasks.

3.2 Chart Metadata

Based on the data collection and auto chart visualization process, we can obtain various chart-related data, including chart table information, chart visualization code, and visual element configurations. These components form the foundational metadata for multi-task chart understanding.

In addition to these, we generate 10 chart-related low-level analysis tasks using local Python code, as shown in Figure 2. These tasks include statistical feature extraction (e.g., mean, median, standard deviation), trend analysis (e.g., increasing, decreasing, stable), and pattern detection, like finding outliers. These low-level analyses provide essential insights into the underlying data, enabling models to perform tasks such as numerical reasoning, trend prediction, and anomaly detection.

Furthermore, we generate chart captions from two dimensions using LLaMA 3.1-70B [7], as shown in Figure 2:❹. The first dimension, chart overview, provides a high-level summary of the chart's content. The second dimension, chart analysis, offers a low-level interpretation of the data, highlighting key insights and patterns. These captions not only enhance the interpretability of the charts but also serve as valuable training data for tasks like chart summarization and question answering.

By combining the generated low-level analysis tasks and multi-dimensional chart captions with the original chart data, we now possess a rich collection of chart metadata that supports multiple downstream chart understanding tasks. This comprehensive metadata enables a wide range of applications, from numerical reasoning and trend analysis to semantic interpretation and summarization, ensuring that our dataset is not only diverse and scalable but also highly practical for real-world chart analysis scenarios.

3.3 An Overview of MetaChart Datasets

We further build a large-scale and high-quality dataset MetaChart using the ChartCards framework. In this section, we provide an overview of existing chart-related training datasets and highlight the advantages of MetaChart based on Table 1. Unlike many existing datasets that focus on single tasks or limited metadata, MetaChart stands out as a comprehensive and versatile dataset designed for multi-task learning. Our automated data generation framework, based on ChartCards, has produced 85K chart-table pairs, 170K chart captions, and 85K visualization codes, making it one of the largest and most diverse datasets in the field. The framework's input,

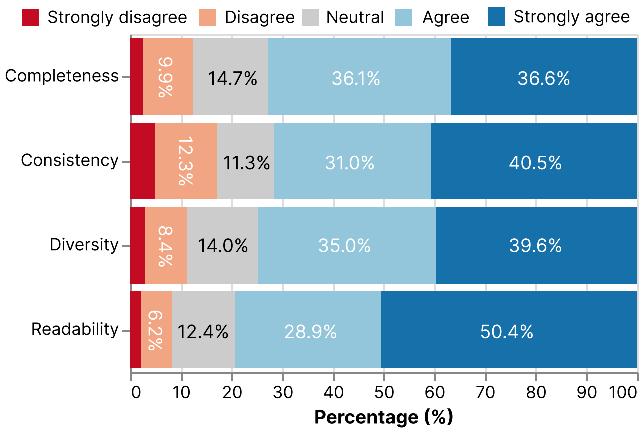

Figure 3: Evaluation of caption quality on four aspects

which relies on CSV files, ensures exceptional scalability, allowing for easy expansion and adaptation to future needs. Additionally, MetaChart incorporates rich metadata, including data tables (DT), visual elements (VE), visualization codes (VC), and chart captions (CC), providing a holistic foundation for model training.

3.4 Crowsourcing Validation on MetaChart

To validate the quality of the generated captions, we conducted a crowdsourcing experiment involving 100 workers who evaluated the captions based on four criteria: completeness, consistency, diversity, and readability. Figure 3 showed that approximately 90% of the workers rated the captions with a score of 3 or higher, indicating high-quality caption generation. This combination of scalability, metadata richness, and high-quality annotations positions MetaChart as a superior dataset for advancing research in chart-related tasks, offering a robust foundation for both current and future applications.

4 Experiments

This section presents how to use MetaChart to help MLLMs improve their capabilities on five downstream chart understanding and reasoning tasks: Text-to-chart retrieval, chart description, chart summary, chart-to-table, and chartQA. For the text-to-retrieval tasks, we employ three retrieval-capable models: CLIIP-DPR [26], UniVL-DR [26], and Long-CLIP [53]. These models are chosen for their ability to map textual queries to visual representations effectively. For the four left generative tasks, we employ three state-of-the-art MLLMs: Qwen2-VL-7B [43], Llama 3.2-11B [7], and LLaVA-NeXT-Mistral-7B [25]. All the training data for the six models for the five tasks are from MetaChart.

4.1 Text-to-chart Retrieval

4.1.1 Test Datasets Overview. We conduct evaluation experiments on the text-to-retrieval task using CRBench [47], which consists of precise queries and fuzzy queries. This benchmark contains 326 queries and 21,862 charts.

4.1.2 Metrics. We evaluate the retrieval performance using five standard metrics, where higher values indicate better performance for all metrics. R@K (K=1,5,10) measures the percentage of queries where the relevant document appears in the top K retrieved results. MRR@10 (Mean Reciprocal Rank at top 10) evaluates the average reciprocal of the rank at which the first relevant document is retrieved. NDCG@10 (Normalized Discounted Cumulative Gain at top 10) measures the ranking quality by considering both the relevance and position of retrieved results.

4.1.3 Training Details. When training text-to-retrieval models, we select CLIP-DPR [8], UniVL-DR [26], and Long-CLIP [53] as base models. Based on the MetaChart datasets we made, we use all chart overview captions co, chart analysis captions ca, and relevant charts c as training sets. During training, we shuffled all the 85K charts with two relevant captions and used cc to represent random captions. Subsequently, we apply the *contrastive learning* based on the *CLIP framework* to optimize the model. The contrastive loss function is defined as:

$$L_i = -\log \frac{\exp^{(sim(c_i, cc_i)/\tau)}}{\sum_{j=1}^{N} \exp^{(sim(c_i, cc_j)/\tau)}}, \quad cc_i, cc_j \in [co, ca]$$

4.1.4 Inference Details. During inference, the model retrieves the most relevant charts from a repository C based on T_i. Then, the query and charts are all encoded into embeddings $\vec{Q}$ and $\vec{C}$, respectively. The similarity between $\vec{Q}$ and $\vec{C}$ is computed using cosine similarity: $\text{sim}(Q, C_i) = \frac{\vec{Q} \cdot \vec{C_i}}{\|\vec{Q}\| \|\vec{C_i}\|}$. The top-$k$ charts with the highest similarity scores are retrieved and presented as the most relevant results.

4.1.5 Overall Results. The experimental results demonstrate consistent and significant improvements across all models after fine-tuning, as shown in Table 2. We can observe that fine-tuning consistently improves performance across all models and metrics. For instance, FT-Long-CLIP-L achieves substantial gains over Long-CLIP-L, with R@10 improving from 79.49% to 84.10% on precise queries and from 74.81% to 80.15% on fuzzy queries. Besides Long-CLIP Under precise queries, FT-CLIP-DPR shows remarkable gains (↑4.62% in R@10), while FT-Univl-DR demonstrates even larger improvements (↑8.71% in R@10). The enhancement is also significant for fuzzy queries, where retrieval is inherently more challenging. For example, FT-Long-CLIP-B substantially improves across all metrics, with NDCG@10 increasing by 9.08%. These results strongly indicate that fine-tuning effectively enhances the models' ability to understand and match precise and fuzzy queries, with notable improvements in handling complex query variations.

4.2 Chart-to-table

4.2.1 Test Datasets Overview. We conduct evaluation experiments on two chart-to-table datasets: ChartQA [34] and ChartVLM [49]. ChartQA-H contains human-annotated chart-table pairs, ChartQA-M consists of machine-generated pairs, and ChartVLM provides a diverse collection of charts from various domains.

4.2.2 Metrics. We evaluate the table extraction performance using two standard metrics, where higher values indicate better performance. Recall measures the proportion of correct data points extracted from the chart and F1 score provides a balanced measure of precision and recall.

Table 2: Experiment results on Text-to-chart retrieval task.

Models	Precise Query					Fuzzy Query				
	R@1	R@5	R@10	MRR@10	NDCG@10	R@1	R@5	R@10	MRR@10	NDCG@10
CLIP-DPR [26]	4.10	15.38	20.00	8.90	11.55	5.34	7.63	10.69	6.60	7.54
FT-CLIP-DPR	8.21	19.49	24.62	13.08	15.80	7.63	14.50	19.08	10.97	12.88
	↑4.11	↑4.11	↑4.62	↑4.18	↑4.25	↑2.29	↑6.87	↑8.39	↑4.37	↑5.34
UniVL-DR [26]	3.08	11.79	17.44	6.59	9.13	5.34	6.87	9.92	6.26	7.11
FT-UniVL-DR	8.21	17.95	26.15	13.25	16.29	12.21	18.32	20.61	15.08	16.42
	↑5.13	↑6.16	↑8.71	↑6.66	↑7.16	↑6.87	↑11.45	↑10.69	↑8.82	↑9.31
Long-CLIP-B [53]	26.67	53.85	62.56	37.99	43.90	22.90	40.46	51.91	30.29	35.33
FT-Long-CLIP-B	38.97	62.05	70.26	49.30	61.30	29.77	50.38	61.83	38.96	44.41
	↑12.30	↑8.20	↑7.70	↑11.31	↑17.40	↑6.87	↑9.92	↑9.92	↑8.67	↑9.08
Long-CLIP-L [53]	41.03	75.90	79.49	55.32	55.32	38.93	67.18	74.81	50.99	56.77
FT-Long-CLIP-L	47.18	80.00	84.10	61.23	66.90	41.98	75.57	80.15	55.31	61.40
	↑6.15	↑4.10	↑4.61	↑5.91	↑11.58	↑3.05	↑8.39	↑5.34	↑4.32	↑4.63

Table 3: Experiment results on Chart-to-table task.

Models	**ChartQA-H**		**ChartQA-M**		**ChartVLM**	
	Recall	F1	Recall	F1	Recall	F1
Qwen2-VL [43]	73.75	82.35	47.74	55.52	60.10	59.79
FT-Qwen2-VL	85.59	84.92	73.65	71.40	72.41	76.05
	↑11.84	↑2.57	↑25.91	↑15.88	↑12.31	↑16.26
Llama 3.2 [7]	63.17	74.58	32.11	42.48	55.73	66.67
FT-Llama 3.2	94.92	94.10	72.13	70.53	83.09	81.42
	↑31.75	↑19.52	↑40.02	↑28.05	↑27.36	↑14.75
LLaVA-NeXT [25]	18.90	26.13	18.51	26.53	15.84	20.03
FT-LLaVA-NeXT	35.46	39.40	22.53	25.52	40.76	42.21
	↑16.56	↑13.27	↑4.02	↑1.01	↑24.92	↑22.18

4.2.3 Training Details. We use 85K charts and 85K chart tables as training data. For chart-to-table tasks, the training objective is to maximize the conditional probability of the ground truth table sequence y given the input x:

$$\max_{\theta} \sum_{(x,y)\in D} \log P_{\theta}(y|x),$$

In this formulation, x represents the input chart c, y denotes the target table sequence table t, D is the training dataset, and θ represents the model parameters. At the token level, the objective is expanded as:

$$\max_{\theta} \sum_{(x,y)\in D} \sum_{t=1}^{T} \log P_{\theta}(y_t|y_{<t}, x),$$

Here, y_t represents the t-th token in the target sequence, with $y_{<t}$ denoting all tokens before position t and T being the length of the target table sequence. This formulation guides the model to learn the generation of accurate outputs by maximizing the probability of predicting the next correct token at each time step.

4.2.4 Inference Details. During inference, the model generates the target table sequence by predicting the next token at each step:

$$\hat{y} = \arg\max_{y} P_{\theta}(y|x).$$

At the token level, the inference process is:

$$\hat{y}_t = \arg\max_{y_t} P_{\theta}(y_t|y_{<t}, x),$$

where $\hat{y}$ represents the complete generated table sequence and $\hat{y}_t$ denotes the predicted token at position t.

4.2.5 Overall Results. The experimental results in Table 3 demonstrate substantial improvements across all models after fine-tuning. The most notable improvement is observed with Llama 3.2, where fine-tuning leads to remarkable gains on ChartQA-H (Recall: from 63.17% to 94.92%, F1: from 74.58% to 94.10%). The improvements are particularly significant on ChartQA-H, with Qwen2-VL improving by 11.84% in Recall and 2.57% in F1, and Llama 3.2 achieving even larger gains of 31.75% in Recall and 19.52% in F1. Even on the more challenging ChartQA-M dataset, fine-tuning demonstrates clear benefits, with FT-Llama 3.2 achieving the best performance (Recall: 72.13%, F1: 70.53%). Among all models, FT-Llama 3.2 consistently achieves the best performance across all datasets, indicating its superior capability in chart understanding and table extraction tasks after fine-tuning.

4.3 Chart Summary

The training and inference formula for the chart summary task is the same as that for the chart-to-table task, except that in chart-to-table, the model's output is a table sequence. In contrast, in chart summary, the model's output is a summary sequence. We use 85K charts and 85K chart overview captions as training data.

4.3.1 Test Datasets Overview. We evaluate the chart summarization performance on the ChartVLM [49] dataset, which provides a comprehensive collection of charts paired with human-written summaries.

4.3.2 Metrics. We evaluate the summary quality using standard text generation metrics, including ROUGE-1 [22], ROUGE-2, and ROUGE-L, which measure the overlap of unigrams, bigrams, and

Table 4: Experiment results on Chart Summary tasks.

Models	ROUGE-1	ROUGE-2	ROUGE-L	Meteor	Overall
Qwen2-VL [43]	37.70	12.61	23.54	19.93	23.95
FT-Qwen2-VL	41.74	12.62	25.14	29.88	27.10
	↑+3.04	↑+0.01	↑+1.60	↑+9.95	↑+3.15
Llama 3.2 [7]	34.12	8.75	19.43	21.78	21.02
FT-Llama 3.2	40.29	11.74	24.76	27.87	26.42
	↑+6.17	↑+2.99	↑+5.33	↑+6.09	↑+5.40
LLaVA-NeXT [25]	41.56	12.94	25.13	30.74	27.59
FT-LLaVA-NeXT	42.47	13.46	26.11	29.65	27.92
	↑+0.91	↑+0.52	↑+0.98	↓-1.09	↑+0.33

longest common subsequences between generated and reference summaries, respectively. We also employ Meteor [2] for additional semantic matching evaluation. The Overall score is calculated as the average of all metrics, with higher values indicating better performance.

4.3.3 Overall Results. The experimental results in Table 4 demonstrate moderate improvements after fine-tuning across most models. The most notable improvements are observed in Llama 3.2, with consistent gains across all metrics (Overall score: from 21.02% to 26.42%, an increase of 5.40%). For Qwen2-VL, significant improvements are seen in Meteor score (↑9.95%) and ROUGE-1 (↑3.04%), leading to an overall improvement of 3.15%. Interestingly, LLaVA-NeXT shows minimal changes after fine-tuning, with slight improvements in ROUGE scores but a small decrease in Meteor (-1.09%), resulting in a marginal overall improvement of 0.02%. These results suggest that while fine-tuning generally enhances summary generation capabilities, the benefits vary across different models and metrics, with Llama 3.2 showing the most consistent and substantial improvements.

4.4 Chart Description

The training and inference formula for the chart description task is the same as that for the chart-to-table task, except that in chart-to-table, the model's output is a table sequence. In contrast, in chart summary, the model's output is a description sequence. We use 85K charts and 85K chart analysis captions as training data.

4.4.1 Test Datasets Overview. We evaluate the chart description generation performance on the ChartVLM [49] dataset, where models are required to generate detailed analytical descriptions of charts.

4.4.2 Metrics. We employ the same set of evaluation metrics as in the summarization task: ROUGE-1 [22], ROUGE-2, and ROUGE-L for measuring n-gram overlap and Meteor [2] for semantic similarity. The Overall score represents the average of all metrics, with higher values indicating better performance.

4.4.3 Overall Results. The experimental results in Table 5 show varied but generally positive improvements after fine-tuning. Llama 3.2 demonstrates consistent improvements across all metrics, with an overall increase of 3.86% (from 26.88% to 30.74%). Notably, LLaVA-NeXT shows substantial improvement in ROUGE-2 (↑12.46%), though with a slight decrease in Meteor (-2.59%), resulting in an overall gain of 2.50%. Qwen2-VL exhibits mixed performance, with decreases in ROUGE-1 (-0.81%) and ROUGE-L (-1.70%), but achieves significant improvement in Meteor (↑7.80%), leading to a positive overall improvement of 2.40%. These results indicate that while fine-tuning generally enhances description generation capabilities, the improvements vary across different aspects of generation quality, with some models showing stronger improvements in semantic similarity (Meteor) while others excel in n-gram matching (ROUGE scores).

Table 5: Experiment results on Chart Description tasks.

Models	ROUGE-1	ROUGE-2	ROUGE-L	Meteor	Overall
Qwen2-VL [43]	44.56	19.53	33.81	29.88	31.45
FT-Qwen2-VL	43.75	19.85	32.11	37.68	33.85
	↓-0.81	↑+0.32	↓-1.70	↑+7.80	↑+2.40
Llama 3.2 [7]	37.55	12.70	27.37	27.91	26.88
FT-Llama 3.2	41.57	16.55	29.20	34.63	30.74
	↑+4.02	↑+3.85	↑+1.83	↑+6.72	↑+3.86
LLaVA-NeXT [25]	39.17	16.90	29.02	36.79	30.72
FT-LLaVA-NeXT	40.97	29.36	29.36	34.20	33.22
	↑+1.80	↑+12.46	↑+0.34	↓-2.59	↑+2.50

Table 6: Zero-shot experiment results on ChartQA tasks.

Models	ChartQA	ChartVLM	Overall
Qwen2-VL [43]	48.50	53.28	50.87
FT-Qwen2-VL	51.50	51.38	51.43
	↑+3.00	↓-1.90	↑+0.56
Llama 3.2 [9]	29.50	39.41	33.81
FT-Llama 3.2	39.50	52.65	44.54
	↑+10.00	↑+13.24	↑+10.73
LLaVA-NeXT [25]	1.50	2.33	1.46
FT-LLaVA-NeXT	29.00	31.36	30.14
	↑+27.50	↑+29.03	↑+28.68

4.5 ChartQA

4.5.1 Test Datasets Overview. We evaluate the zero-shot chart question answering performance on two datasets, ChartQA [34] and ChartVLM [49].

4.5.2 Metrics. We evaluate the chart question answering performance using accuracy scores, where higher values indicate better performance. The Overall score is calculated as the average accuracy across all three datasets.

4.5.3 Inference Details. For the Chart QA task, we directly utilize the model fine-tuned on the chart-to-table task without additional training. The inference process consists of two steps. First, we generate the table representation from the chart:

$$\hat{t} = \arg\max_{t} P_{\theta}(t|c)$$

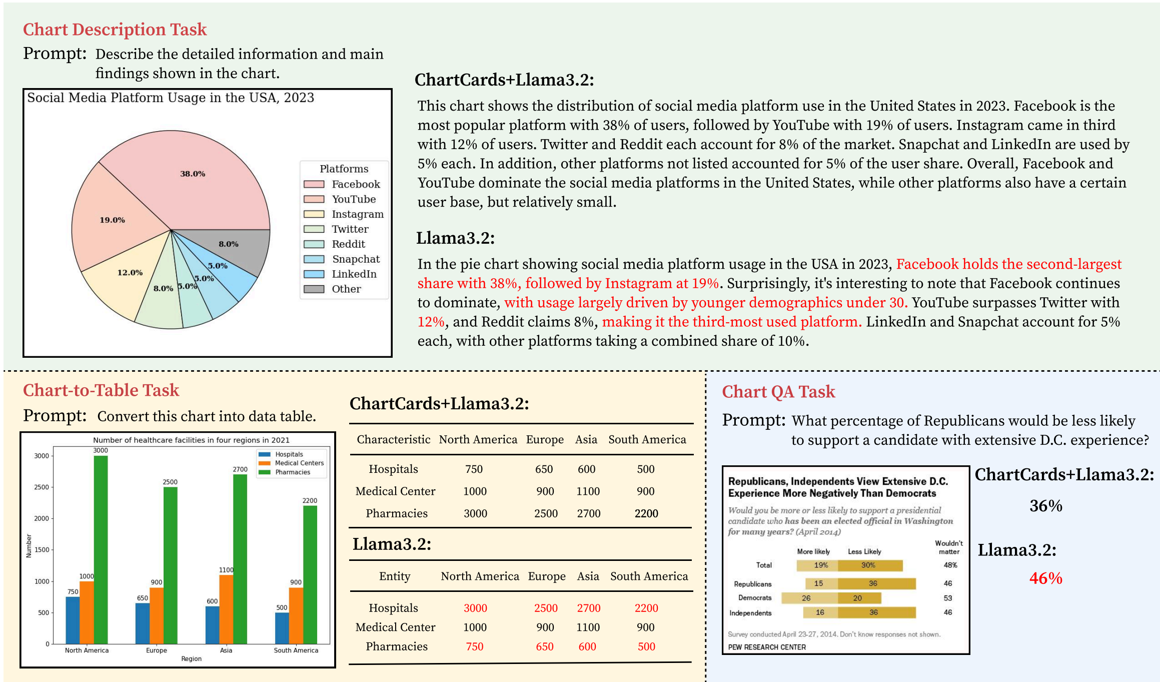

Figure 4: Case study demonstrating the effectiveness of MetaChart in improving MLLM understanding and performance across various chart analysis tasks.

Then, we use both the chart and its generated table as input to verify the quality of the dataset. The inference process can be formulated as:

$$\hat{a} = \arg\max_{a} P_{\theta}(a|c, \hat{t})$$

where $\hat{a}$ represents the generated answer, c is the input chart image, $\hat{t}$ is the generated table, and $\hat{a}$ is the predicted answer. This zero-shot inference approach leverages the table-generation capabilities of the model while incorporating additional prompt information to guide the answer-generation process.

4.5.4 Overall Results. The experimental results in Table 6 demonstrate varying degrees of improvement after fine-tuning across different models. LLaVA-NeXT shows the most dramatic improvements, with substantial gains across all datasets (Overall: from 1.46% to 30.14%, ↑28.68%) . Llama 3.2 demonstrates consistent improvements across all datasets, achieving an overall improvement of 10.73%, with the strongest gain on ChartVLM (↑13.24%). In contrast, Qwen2-VL shows mixed results, with improvement on ChartQA (↑3.00%) but slight decreases on ChartVLM (-1.90%), leading to a marginal overall improvement of 0.56%. These results suggest that while fine-tuning generally enhances zero-shot capabilities, the benefits vary significantly across different models and datasets.

4.6 Qualitative Results.

As shown in Figure 4, we demonstrate the performance of MetaChart data augmentation when fine-tuned with Llama 3.2 [7] for various chart analysis tasks, revealing how fine-tuning with MetaChart addresses critical gaps in structural and contextual understanding. In the Chart-to-Description task, the original model conflates color proximity and label positioning, and even generates hallucinations (*e.g.*, younger demographics) not supported by visual cues. After fine-tuning, the model integrates multi-modal context (*e.g.*, label-text alignment) to resolve color confusion while filtering out irrelevant text noise. In the Chart-to-Table task, the original model fails to effectively convert data into a structured table, misrepresenting key values and failing to correctly organize the data due to issues with structural parsing. The fine-tuned model accurately handles the relationships between different categories (*e.g.*, hospitals and other regions), which the original model struggled with. In the chartQA task, after data augmentation with MetaChart, the fine-tuned model not only parses explicit data from the chart but also combines contextual semantics, logical reasoning, and noise resistance to accurately answer complex questions. These results highlight how MetaChart data augmentation enhances the model's ability to correctly process and interpret visual information, significantly improving performance in tasks such as chart analysis, table generation, and reasoning.

5 Conclusion

In this paper, we introduce ChartCards, a chart-metadata generation framework for multi-task chart understanding. Based on this framework, we constructed the MetaChart dataset, which offers a richer selection of chart metadata and higher data quality than existing datasets of its kind. Experimental results demonstrate that six models fine-tuned on MetaChart achieve notable performance improvements, further validating the effectiveness and quality of the ChartCards framework.

References

[1] 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 (2022), 23716–23736.

[2] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss (Eds.). Association for Computational Linguistics, Ann Arbor, Michigan, 65–72. https://aclanthology.org/W05-0909/

[3] Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Zhiqing Sun, Dan Gutfreund, and Chuang Gan. 2024. Visual chain-of-thought prompting for knowledge-based visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 1254–1262.

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

[5] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research* 25, 70 (2024), 1–53.

[6] Kenny Davila, Srirangaraj Setlur, David Doermann, Bhargava Urala Kota, and Venu Govindaraju. 2021. Chart Mining: A Survey of Methods for Automated Chart Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 11 (2021), 3799–3819. doi:10.1109/TPAMI.2020.2992028

[7] Aaron Grattafiori, Abhimanyu Dubey, Zhiwei Zhao et al., and Zhiyu Ma. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[8] Markus Hafner, Maria Katsantoni, Tino Köster, James Marks, Joyita Mukherjee, Dorothee Staiger, Jernej Ule, and Mihaela Zavolan. 2021. CLIP and complementary methods. *Nature Reviews Methods Primers* 1, 1 (2021), 20.

[9] 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* (2023).

[10] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.

[11] Linmei Hu, Duokang Wang, Yiming Pan, Jifan Yu, Yingxia Shao, Chong Feng, and Liqiang Nie. 2024. NovaChart: A Large-scale Dataset towards Chart Understanding and Generation of Multimodal Large Language Models. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 3917–3925.

[12] 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*. 5648–5656.

[13] Kaggle. 2025. Datasets. https://www.kaggle.com/datasets

[14] 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* (2017).

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

[16] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691* (2021).

[17] Dongxu Li, Junnan Li, Hung Le, Guangsen Wang, Silvio Savarese, and Steven CH Hoi. 2022. Lavis: A library for language-vision intelligence. *arXiv preprint arXiv:2209.09019* (2022).

[18] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, 19730–19742.

[19] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems* 34 (2021), 9694–9705.

[20] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557* (2019).

[21] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. 2022. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10965–10975.

[22] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013/

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

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

[25] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/

[26] Zhenghao Liu, Chenyan Xiong, Yuanhuiyi Lv, Zhiyuan Liu, and Ge Yu. 2022. Universal vision-language dense retrieval: Learning a unified representation space for multi-modal retrieval. *arXiv preprint arXiv:2209.00179* (2022).

[27] 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 (2019).

[28] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255* (2023).

[29] Yuyu Luo, Chengliang Chai, Xuedi Qin, Nan Tang, and Guoliang Li. 2020. Interactive Cleaning for Progressive Visualization through Composite Questions. In *ICDE*. IEEE, 733–744.

[30] Yuyu Luo, Xuedi Qin, Nan Tang, and Guoliang Li. 2018. DeepEye: Towards Automatic Data Visualization. In *ICDE*. IEEE Computer Society, 101–112.

[31] Yuyu Luo, Xuedi Qin, Nan Tang, Guoliang Li, and Xinran Wang. 2018. DeepEye: Creating Good Data Visualizations by Keyword Search. In *SIGMOD Conference*. ACM, 1733–1736.

[32] Yuyu Luo, Yihui Zhou, Nan Tang, Guoliang Li, Chengliang Chai, and Leixian Shen. 2023. Learned Data-aware Image Representations of Line Charts for Similarity Search. In *Proceedings of the ACM on Management of Data, SIGMOD'23*. ACM, 1–29. doi:10.1145/3588942

[33] 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*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). ACL, Singapore, 14662–14684. doi:10.18653/v1/2023.emnlp-main.906

[34] 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* (2022).

[35] 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*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). ACL, Bangkok, Thailand, 7775–7803. doi:10.18653/v1/2024.findings-acl.463

[36] 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*. 1527–1536.

[37] OpenAI. 2024. Introducing OpenAI o1-preview. https://openai.com/index/introducing-openai-o1-preview/

[38] Leixian Shen, Enya Shen, Yuyu Luo, Xiaocong Yang, Xuming Hu, Xiongshuai Zhang, Zhiwei Tai, and Jianmin Wang. 2023. Towards Natural Language Interfaces for Data Visualization: A Survey. *IEEE Transactions on Visualization and Computer Graphics* 29, 6 (2023), 3121–3144. doi:10.1109/TVCG.2022.3148007

[39] Shuyu Shen, Sirong Lu, Leixian Shen, Zhonghua Sheng, Nan Tang, and Yuyu Luo. 2025. Ask Humans or AI ? Exploring Their Roles in Visualization Troubleshooting. *arXiv* (2025).

[40] Chufan Shi, Cheng Yang, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, et al. 2024. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation. *arXiv preprint arXiv:2406.09961* (2024).

[41] Benny J Tang, Angie Boggust, and Arvind Satyanarayan. 2023. Vistext: A benchmark for semantically rich chart captioning. *arXiv preprint arXiv:2307.05356* (2023).

[42] Yuan Tian, Weiwei Cui, Dazhen Deng, Xinjing Yi, Yurun Yang, Haidong Zhang, and Yingcai Wu. 2024. Chartgpt: Leveraging llms to generate charts from abstract natural language. *IEEE Transactions on Visualization and Computer Graphics* (2024).

[43] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jingren Zhou Jinze Bai et al., and Junyang Lin. 2024. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. arXiv:2409.12191 [cs.CV] https://arxiv.org/abs/2409.12191

[44] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. 2023. Seggpt: Segmenting everything in context. *arXiv preprint arXiv:2304.03284* (2023).

[45] Yang Wu, Yao Wan, Hongyu Zhang, Yulei Sui, Wucai Wei, Wei Zhao, Guandong Xu, and Hai Jin. 2024. Automated Data Visualization from Natural Language via Large Language Models: An Exploratory Study. In *Proceedings of the ACM on*

Management of Data. ACM, 1–28. doi:10.1145/3654992

[46] Yifan Wu, Lutao Yan, Leixian Shen, Yunhai Wang, Nan Tang, and Yuyu Luo. 2024. ChartInsights: Evaluating Multimodal Large Language Models for Low-Level Chart Question Answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. ACL, Stroudsburg, PA, USA, 12174–12200. doi:10.18653/v1/2024.findings-emnlp.710

[47] Yifan Wu, Lutao Yan, Yizhang Zhu, Ning Liu, Yinan Mei, Jiannan Wang, Nan Tang, and Yuyu Luo. 2025. Boosting Text-to-Chart Retrieval with Semantic Insights. *arXiv preprint* (2025).

[48] Renqiu Xia, Bo Zhang, Haoyang Peng, Hancheng Ye, Xiangchao Yan, Peng Ye, Botian Shi, Yu Qiao, and Junchi Yan. 2023. Structchart: Perception, structuring, reasoning for visual chart understanding. *arXiv preprint arXiv:2309.11268* (2023).

[49] Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Min Dou, Botian Shi, Junchi Yan, et al. 2024. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *arXiv preprint arXiv:2402.12185* (2024).

[50] Zhengzhuo Xu, Sinan Du, Yiyan Qi, Chengjin Xu, Chun Yuan, and Jian Guo. 2023. Chartbench: A benchmark for complex visual reasoning in charts. *arXiv preprint arXiv:2312.15915* (2023).

[51] Xudong Yang, Yifan Wu, Yizhang Zhu, Nan Tang, and Yuyu Luo. 2024. AskChart: Universal Chart Understanding through Textual Enhancement. *CoRR* abs/2412.19146 (2024).

[52] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917* (2022).

[53] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. 2024. Long-clip: Unlocking the long-text capability of clip. In *European Conference on Computer Vision*. Springer, 310–325.

[54] Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. 2024. Tinychart: Efficient chart understanding with visual token merging and program-of-thoughts learning. *arXiv preprint arXiv:2404.16635* (2024).

[55] Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Wanxiang Che, Zhiyuan Liu, and Maosong Sun. 2025. ChartCoder: Advancing Multimodal Large Language Model for Chart-to-Code Generation. *arXiv preprint arXiv:2501.06598* (2025).