EvoChart: A Benchmark and a Self-Training Approach Towards Real-World Chart Understanding

Muye Huang[1,2], **Han Lai**[1,2], **Xinyu Zhang**[1,3], **Wenjun Wu**[1,3], **Jie Ma**[2*], **Lingling Zhang**[1,2], **Jun Liu**[1,2]

[1]School of Computer Science and Technology, Xi'an Jiaotong University
[2]MOE KLINNS Lab, Xi'an Jiaotong University
[3]Shaanxi Province Key Laboratory of Big Data Knowledge Engineering
{huangmuye, hanlai, zhang1393869716, nickjun98}@stu.xjtu.edu.cn,
{jiema, zhanglling, liukeen}@xjtu.edu.cn

Abstract

Chart understanding enables automated data analysis for humans, which requires models to achieve highly accurate visual comprehension. While existing Visual Language Models (VLMs) have shown progress in chart understanding, the lack of high-quality training data and comprehensive evaluation benchmarks hinders VLM chart comprehension. In this paper, we introduce EvoChart, a novel self-training method for generating synthetic chart data to enhance VLMs' capabilities in real-world chart comprehension. We also propose EvoChart-QA, a noval benchmark for measuring models' chart comprehension abilities in real-world scenarios. Specifically, EvoChart is a unique self-training data synthesis approach that simultaneously produces high-quality training corpus and a high-performance chart understanding model. EvoChart-QA consists of 650 distinct real-world charts collected from 140 different websites and 1,250 expert-curated questions that focus on chart understanding. Experimental results on various open-source and proprietary VLMs tested on EvoChart-QA demonstrate that even the best proprietary model, GPT-4o, achieves only 49.8% accuracy. Moreover, the EvoChart method significantly boosts the performance of open-source VLMs on real-world chart understanding tasks, achieving 54.2% accuracy on EvoChart-QA.

Homepage — https://github.com/MuyeHuang/EvoChart

1 Introduction

Chart Question Answering (CQA) aims to answer specific questions based on the context provided by chart images, enabling automated data analysis, such as the business data reports. This process requires complex chart understanding and visual reasoning skills to interpret various elements, including visual components, text and values. Consequently, CQA tasks have attracted the interest of researchers (Kafle et al. 2018; Methani et al. 2020; Masry et al. 2022).

Recently, VLMs (Liu et al. 2023b; Dai et al. 2023; Lin et al. 2023; Zhu et al. 2024) have shown significant advancements in general visual capabilities, especially in chart understanding, achieving high scores on the ChartQA dataset (Zhang et al. 2024; Chen et al. 2023a; Meng et al. 2024).

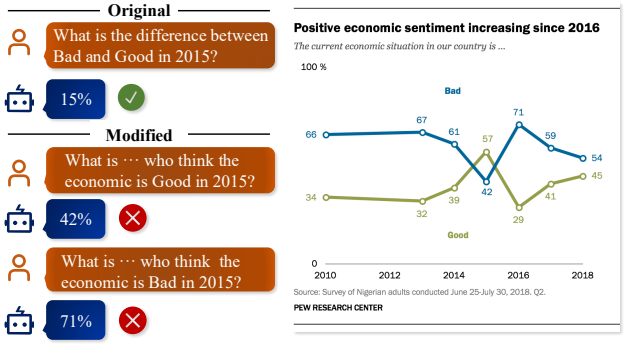

Figure 1: Case of Modified ChartQA. "Original" refers to the question from the ChartQA dataset, while "Modified" refers to our modified version.

However, their real-world performance is notably weaker than their ChartQA test set performance. We conducted a test to illustrate this, as shown in Figure 1, we discarded complex reasoning problems in the ChartQA (Masry et al. 2022) training set and posed 103 basic understanding questions. We then evaluated various VLMs on these questions. The results, presented in the Appendix, show that performance dropped by over 40% compared to ChartQA scores, even for questions on training set charts. This highlights two points: first, current VLMs are capable of answering some chart-reasoning questions, but they lack a comprehensive understanding of charts. Second, the ChartQA dataset allows models to correctly answer questions without a comprehensive understanding of the charts, leading to an overestimation of the capabilities of current models (Wang et al. 2024).

Firstly, the lack of high-quality chart training data is a major reason why current models lack robust chart understanding capabilities. Existing methods (Masry et al. 2022, 2023; Meng et al. 2024) for collecting chart training data fall into two categories: manual annotation and automatic synthesis. Manually annotated data have real-world chart appearances but suffer from coarse granularity and high human costs. Automatically synthesized data offer fine-grained annotations but lack real-world diversity, leading to poor model robustness. Thus, constructing chart datasets is a challenging bal-

*Corresponding author.

ance between cost and quality, resulting in a scarcity of high-quality chart training data.

Secondly, the single source of charts and the excessive focus on high-level chart reasoning are primary reasons why the ChartQA dataset provides an overly optimistic estimation of VLM chart understanding capabilities. The ChartQA dataset has only four chart sources, focus on politics and economics. Each source of charts with similar styles, making it prone to overfitting. Additionally, datasets like ChartQA focuses heavily on numerical and logical reasoning, this allows the model to potentially answer questions correctly without a clear understanding of the chart. For example, "*What is the difference between Bad and Good in 2015?*", the model may not explicitly know the values of "Good" and "Bad" in 2015, but still has the possibility of answering the question accurately.

To address these challenges, we propose a novel method, EvoChart, for synthesizing high-quality chart datasets with real-world characteristics. We also introduce EvoChart-QA, a carefully crafted benchmark for evaluating chart comprehension in real-world scenarios. EvoChart is a multi-stage self-training approach for chart data generation. In each stage, the chart generator produces a batch of synthetic chart data, and the model self-selects and refines the chart data, ensuring that the synthesized data is of high quality for current stage. Subsequently, the model trains on the self-selected data to progress to the next stage. This approach produces both a progressively challenging dataset and a robust chart understanding model. EvoChart-QA is a benchmark designed for basic chart understanding, featuring 650 charts from 140 real-world websites and 1250 expert-curated questions. The diverse chart styles accurately simulate real-world scenarios, with questions focused on chart understanding. Experiments on EvoChart-QA demonstrate that our EvoChart method achieves outstanding performance with 54.2% accuracy, also exhibits leading performance of 81.5% on the ChartQA dataset.

Our main contributions are summarized into three folds:

- We propose EvoChart, a method that combines chart dataset construction with model self-training, using a multi-stage approach to simultaneously output high-quality chart data and a chart understanding model.
- We propose a novel real-world chart basic understanding benchmark, EvoChart-QA, which comprehensively evaluates a model's chart understanding capability through multi-source real-world charts and multi-type manually curated questions.
- We conducted extensive experiments on the EvoChart method and EvoChart-QA. Results demonstrate that the EvoChart method significantly outperforms other data synthesis methods, and we also deeply analyze the performance of various VLMs on EvoChart-QA.

2 Related Work

2.1 Chart Question Answering Datasets

Since FigureQA (Kahou et al. 2018) pioneered the CQA task, numerous datasets for chart question answering have emerged. Synthetic datasets, such as DVQA (Kafle et al. 2018), PlotQA (Methani et al. 2020), RealCQA (Ahmed et al. 2023), ChartX (Xia et al. 2024) and UniChart (Masry et al. 2023). This datasets utilize synthetically generated charts or templat-base questions. Generate datasets such as ChartSFT (Meng et al. 2024), utilize a mixture of GPT-4 (OpenAI et al. 2024)-generated charts and questions. Mixed datasets as ChartQA (Masry et al. 2022) and Charxiv (Wang et al. 2024), the former is a dataset compiled semi-manually with the assistance of templates, while the latter is template-based and requires evaluation by GPT-4o. In contrast, EvoChart-QA focus on real-world scenarios and employ an automated evaluation method that does not necessitate the utilization of GPT-4.

2.2 Visual Language Models on CQA

VLMs are language models with visual understanding capabilities, and they have numerous applications in CQA tasks. Small VLMs like ChartReader (Cheng, Dai, and Hauptmann 2023), MatCha (Liu et al. 2023a), ScreenAI (Baechler et al. 2024) and UniChart (Masry et al. 2023) have shown superior performance on tasks like PlotQA and DVQA, highlighting the potential of VLMs in CQA. ChartLlama (Han et al. 2023) was a milestone, being the first to apply LLaVa1.5 (Liu et al. 2024) to CQA tasks and achieving impressive performance. Subsequently, works such as ChartPaLI (Carbune et al. 2024), ChartInstruct (Masry et al. 2024a), ChartAst-D (Meng et al. 2024), and TinyChart (Zhang et al. 2024) delved into the multimodal alignment and CQA reasoning aspects of VLMs in CQA, achieving remarkable performance. Recently, open-source general VLMs such as Phi3-Vision (Abdin et al. 2024) and Intern-VL2.0 (Chen et al. 2023b), through large-scale training, have achieved state-of-the-art performance on the ChartQA dataset.

2.3 Self-Training Approach

With the increasing capabilities of language models, numerous researchers have begun to explore the potential of leveraging language models for self-training. GPT3Mix (Yoo et al. 2021) proved that large language model augmentation of textual corpora is very effective. Further, ReST (Gülçehre et al. 2023) achieves cost-effective and efficient human preference alignment through a dual-loop self-training approach. Dennis et al. (Ulmer et al. 2024) obtained new data from the self-talk of multi-role-playing LLM Agents by adding a filtering check mechanism, realizing efficient self-training. Recently, Xu et al. (Xu et al. 2024) proposed ENVISIONS, which uses a neural-symbolic self-training approach to significantly improve mathematical and logical reasoning abilities without relying on external stronger models or evaluation tools. Inspired by their work, our proposed EvoChart focuses on a scalable self-training process.

3 EvoChart Method

We introduce the EvoChart, a unique self-training data synthesis approach that simultaneously produces high-quality training corpus and a high-performance chart understanding

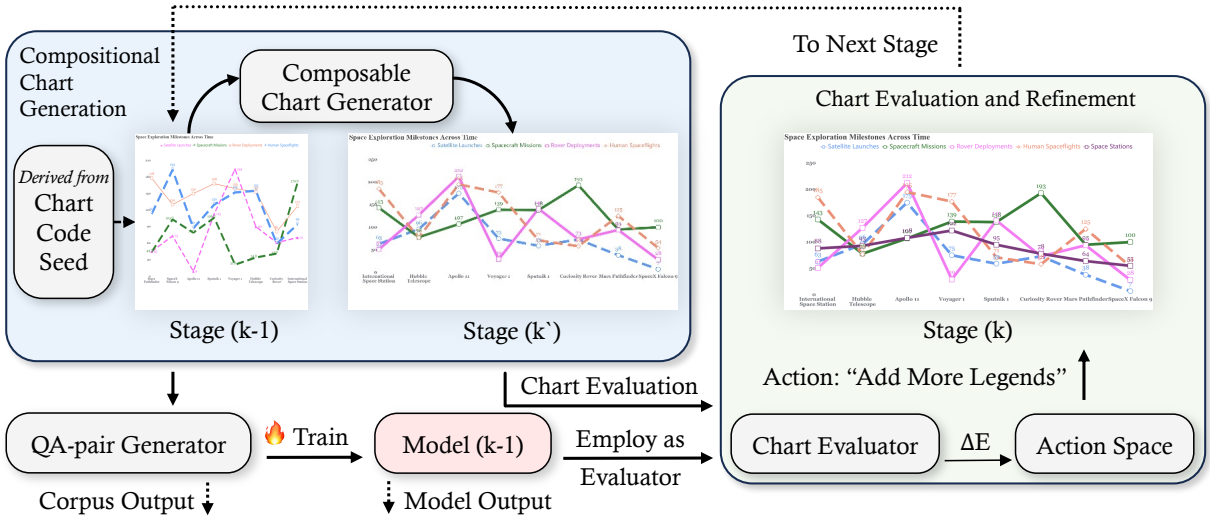

Figure 2: The overview of the proposed EvoChart method. The figure depicts a counterclockwise cyclical self-training process, where the Chart Evaluator of each stage k is trained based on the results of the previous stage $k-1$.

model. EvoChart comprises three iterative phases: Compositional Chart Generation for generating charts with diverse appearance, Chart Evaluation and Refinement to select and refine charts suitable for the current stage, and QA-pair generation and training to produce training data and provide a stronger model for the subsequent stage. These phases operate cyclically throughout the construction of EvoChart, as illustrated in Figure 2. We will explore each of these steps in the following subsections.

3.1 Compositional Chart Generation

Compositional chart generation aims to produce high-quality and diverse charts at minimal cost, serving as the core component of EvoChart's construction. Previous approaches (Han et al. 2023; Meng et al. 2024; Zhang et al. 2024) either rely solely on GPT-4 for chart generation result in limited diversity and high costs, or using plotting libraries for random generation often leads to unrealistic themes and styles. To achieve continuous, low-cost, and diverse chart creation, we propose a two-step generation strategy within the compositional chart generation process:

1) Chart Code Seed Generation: This step serves as the initial phase of chart generation, aiming to produce fundamental chart code containing elements that are difficult to achieve via random processes. These elements include chart themes and appropriate units for the x- and y-axes. Notably, this step is executed only once during the entire chart dataset construction process. As (Xu et al. 2023) demonstrated the feasibility of large language models for code generation, we employ sophisticated prompt engineering techniques to guide GPT-4 in generating over 25k real-world chart code seeds. Our GPT-4 prompts are based on the following key aspects:

Chart Types: Different chart types are suited for different themes. We focus on the four most prevalent real-world chart types: line charts, bar charts, pie charts, and scatter charts. For each of these chart types, we generate themes that are specifically tailored to their characteristics and use cases.

Chart Themes: Chart themes are challenging to generate through any random process, and manual curation is both costly and prone to domain bias. By utilizing prompting GPT-4, we have generated 25,000 themes across over 200 domains, including politics, economics, technology and everyday life. These themes encompass various titles, units, and other relevant elements.

Chart Color Schemes: Chart color schemes significantly influence the visual appearance of a chart. We employ an automated approach to generate over 200 color palettes, ensuring aesthetically pleasing chart appearances. These color schemes include diverse colors for lines, bars, segments, and backgrounds, among other visual elements.

2) Composable Chart Generator: The Composable Chart Generator is responsible for producing diverse charts. It is invoked multiple times during the construction process. This step automates the creation of a wide variety of charts by randomly assigning configurations to the chart code. We have defined over a hundred configuration options, each with dozens of potential values. The generator automatically selects these options based on the Code Seed, ensuring diverse chart outputs. Due to the numerous configuration options, we will highlight a few key aspects below:

Chart Data: Numerical data constitutes the core message conveyed by a chart. While randomly generated data may result in excessively volatile values and unrealistic visualizations, we ensure that the generated chart data is adhered to the specified ranges provided in the Code Seed. This ensures that data values remain within the reasonable bounds defined by the chosen theme.

Axis Tick Interval: Real-world chart creators often omit some labels on axes. For any continuous axis label (e.g.,

year, month, quarter), we set a 25% probability of no omission, a 50% probability of omitting one out of three labels, and a 25% probability of omitting two out of four labels.

Other Configurations: A multitude of detailed configurations influence chart appearance, including line width, numeric label (position), line style (solid or dashed), bar stacking, axis visibility, font size, font type, and more. We introduce randomness into these configurations through a range of selectable options, and we employ ECharts (Li et al. 2018) for rendering charts.

3.2 Chart Evaluation and Refinement

Chart Evaluation and Refinement enhances the chart images generated by the Compositional Chart Generation process. While chart code seeds can produce diverse charts, refinement remains crucial due to the following reasons: 1) Random generation can lead to visually similar charts, causing overfitting and reducing the model's generalization ability. 2) Seed-based chart construction may result in poor chart aesthetics, negatively impacting data quality. To address these issues, we propose two steps: the Chart Evaluator and the Action Space. In the k-th stage of data synthesis (Stage-k), the Chart Evaluator assigns a multi-dimensional evaluation score e_k to the charts. Based on the difference ΔE between e_k and the previous score e_{k-1}, the Action Space selects an action to modify the charts. The detailed process is described below.

The Chart Evaluator uses the current stage model to assess chart quality, producing a multi-dimensional evaluation score. To avoid hallucinations from questions like "*Does this chart have flaws?*", we assess quality using a directness-based question-answering approach. The Action Space then selects actions to refine the charts based on the evaluation scores. A detailed list of action types is in the Appendix. The evaluation questions and actions are as follows:

Is-Chart & Is-Title-Clear: These questions check if the chart is correctly rendered. While existing VLMs struggle to comprehend charts in detail, they can still distinguish the names of different chart types. Therefore, we propose the following questions. For example, "*Is the image a horizontal bar chart?*" If the model answers incorrectly, the action is "Drop." If correct, the action is "None."

Label-Value & Value-Label: This question type evaluates chart quality by examining text-value alignment. For example, "*What is the value of Medication in May?*" We generate 10 questions per chart and calculate the average accuracy e_k. If $e_k - e_{k-1}$ is significantly positive, the chart may be too simple, prompting a "value enhancement method." If significantly negative, it may indicate errors or overlaps, prompting the "Drop" action.

Label-Visual & Visual-Label: This evaluates visual-text alignment, for example, "*What is the bar color of Medication?*" We generate 10 questions per chart and calculate accuracy e_k. If $e_k - e_{k-1}$ is significantly positive, the chart's visual information may be too simple, prompting a "visual enhancement method." If significantly negative, it may indicate visual errors, prompting the "Drop" action.

Through Chart Evaluation and Refinement, we ensure that EvoChart generates accurate and challenging data relative to the current stage model in each stage. This ensures data diversity and simultaneously prevents the EvoChart model from overfitting to the EvoChart Corpus.

3.3 QA-pairs Generation and Training

QA-pairs Generation and Training aims to generate chart-based question-answer pairs, incorporating these data into the EvoChart corpus and training the EvoChart model for the next stage. We generate question-answer pairs using various question templates. Notably, we focus on basic chart understanding, the templates specifically focus on the alignment of visual-text-value information in charts (e.g., extracting values through visual information, extracting visual information through text). Additionally, we generate rich question-CoT pairs using composable vCoTs (Rose et al. 2024) and distinguish Direct from vCoT using Instruct. Since vCoTs solely serve to enhance model comprehension, we only mix in 20% of vCoT data in the training data. We have established 198 question templates with corresponding answers including Direct and over 500 vCoT templates. We generated 1.6M QA-pairs during the training in 3 stages. A detailed information can be found in the Appendix.

4 EvoChart-QA Benchmark

EvoChart-QA is a comprehensive and challenging benchmark for real-world chart understanding. We carefully selected 625 charts with diverse appearances, all sourced from real-world websites. Then we curated 1250 chart-based understanding questions through human experts. This process ensures that EvoChart-QA accurately reflects real-world scenarios. The comparison between EvoChart-QA and other benchmarks is shown in Table 1. In the following sections, we will elaborate on the chart selection process, question construction methods, and evaluation metrics used.

Name	Real Data	Real Chart	Open Vocab	Human Query	Multi Souce	Flex Eval
FigureQA	✗	✗	✗	✗	✗	✗
DVQA	✗	✗	✗	✗	✗	✗
PlotQA	✓	✗	✓	✗	✗	✗
ChartQA	✓	✓	✓	✓	✗	✗
CharXiv	✓	✓	✓	✓	✗	✗
Ours	✓	✓	✓	✓	✓	✓

Table 1: Comparison with different benchmarks

4.1 Chart Selection

To enable EvoChart-QA to emulate real-world chart understanding scenarios, all charts in our dataset are carefully selected by human experts. Specifically, we crawled 1,000 charts from 140 different websites. Human experts then filtered out images with ambiguous meanings or damages, resulting in a final dataset of 625 valid images. These images include line charts, bar charts, pie charts, and scatter plots. Examples and sources of all images are detailed in the Appendix.

4.2 Question Construction

We focus on chart basic understanding questions. Following the definitions of prior researchers (Kafle et al. 2020; Methani et al. 2020), we concentrate on data and structural retrieval questions. Specifically, we categorize the problems into two types during manual construction: Direct Retrieval and Complex Retrieval. Direct Retrieval questions focus on understanding the image and directly extracting its content, while Complex Retrieval questions emphasize performing multiple visual reasoning steps on the chart.

1) Direct Retrieval. Direct Retrieval aims to directly extract elements from a chart based on the question. To comprehensively assess the model's ability to extract various elements from charts, we categorize chart elements into three types: label, value, and visual. Label elements refer to textual content in the chart, such as chart title, axis labels, etc. Value elements refer to data values conveyed by the chart, which are displayed or implicitly provided based on the chart author's intention. Visual elements refer to all visual descriptions in the chart, such as line color, largest segment, etc. For example: "*What is the value of the green dashed line in 2015?*" Although Direct Retrieval involves only extracting elements from charts, it remains a challenging task. On the one hand, real-world charts often exhibit non-standard variations. For example, they may include rich text such as numerous logos inserted in the image, or they may combine multiple chart forms within a single chart to facilitate expressions. On the other hand, questions posed by real-world users may contain ambiguous expressions. For example, "*Which country's total GDP is represented by the bar in the lightest shade of blue?*" This visual description is ambiguous, but it is a clearly question for human observers.

2) Complex Retrieval. Complex Retrieval involves querying information within a chart using complex, multi-step descriptions. Compared to Direct Retrieval, Complex Retrieval focuses on understanding the relative positions of elements within the chart. For example, "*In the chart, the third bar to the left of the longest red bar represents the GDP of which country?*". Complex retrieval poses novel challenges for chart comprehension. This is because the descriptive information in complex retrieval relies entirely on the chart itself, requiring the model to have a comprehensive and clear understanding of the chart. For example, comprehending the previously mentioned "*the longest red bar*" is entirely based on the extraction of information from the chart's content. Furthermore, this also necessitates the model to possess sophisticated visual reasoning capabilities, such as understanding "*the third bar from the left*" which demands visually-grounded inference.

4.3 Evaluation Metrics

We designed a automatic evaluation method for EvoChart-QA, combining flex and strict approaches, to fairly evaluate answer correctness. In EvoChart-QA construction, we label questions as "Strict" or "Flex" and use the corresponding Strict or Flex approach to evaluate correctness. For "Strict" type questions, answers have a definite value, such as numerical or textual values explicitly labeled in the chart. We employ a zero-tolerance approach for judging these questions. For "Flex" type questions, answers have estimated values, such as unlabeled numerical values. We employ a 5% tolerance approach to judge these questions. Finally, we employ average accuracy to evaluate the model's performance. In contrast to our metrics, previous methods like ChartQA allowed a 5% tolerance for any numerical answer, leading to an optimistic estimation of model outputs. For example, years are numerical answers, and 1995 and 2008 would fall within the 5% tolerance in previous evaluation metrics, and our method does not exhibit this error.

5 Experiments

5.1 Setup

Datasets. To comprehensively evaluate the effectiveness of EvoChart, we chose to test it on both ChartQA and EvoChart-QA. ChartQA (Masry et al. 2022) is a dataset with two subsets: "Augment", which is machine-generated, and "Human", which is manually curated. "Augment" focuses on element extraction tasks within machine-synthesized images, while "Human" emphasizes complex numerical and logical reasoning tasks in real-world charts. EvoChart-QA is a novel real-world benchmark that we proposed.

Models. We conducted extensive evaluations on both open-source and proprietary models. For open-source models, we tested Phi3-Vision-4B (Abdin et al. 2024), QwenVL-Chat-7B (Bai et al. 2023), LlaVa1.6-Vicuna-7B (Liu et al. 2024), Intern-VL-2.0-8B (Chen et al. 2023b), Llama3-Llava-Next-8B (Li et al. 2024), CogVLM2-19B (Wang et al. 2023), LlaVa1.6-YI-34B, Intern-VL-2.0-40B, ChartLlama-13B (Han et al. 2023), ChartAst-S-13B (Meng et al. 2024), ChartIns-Llama2-7B (Masry et al. 2024a), ChartIns-FlanT5-3B, ChartGemma-2B (Masry et al. 2024b), and TinyChart-3B (Zhang et al. 2024). For proprietary models, we tested Gemini-1.5-Flash (Team et al. 2024), Gemini-1.5-Pro, Qwen-VL-Plus, Qwen-VL-Max, GPT-4-turbo (OpenAI et al. 2024), and GPT-4o. For all models, we employed a zero-shot approach. The specific configurations of all models are provided in the Appendix.

Settings. In EvoChart method, we utilize Phi3-Vision (Abdin et al. 2024) as the initialization model. We conducted a 3-Stage data synthesis and training process, with each Stage undergoing 1 Epoch of SFT with a learning rate of 2e-5 and using cosine learning rate scheduler. All experiments were completed on 4 NVIDIA A800 80G GPUs.

5.2 Experimental Results

Tables 2 and 3 present the performance of EvoChart and other open-source or proprietary VLMs on EvoChart-QA and ChartQA. We elaborate on the experimental results in two aspects: EvoChart-QA and EvoChart.

EvoChart-QA Results. All models exhibit relatively poor performance on EvoChart-QA, with accuracies not exceeding 55%. Among proprietary models, GPT-4o achieves the highest accuracy at 49.8%. InternVL-2.0-40B demonstrates the strongest performance among open-source general-purpose models, reaching 49.0%. Within the domain of chart-expert models, our proposed EvoChart method yields the best-performing model, achieving the highest accuracy

Model	Line			Bar			Pie			Scatter			Overall		
	Dir.	Comp.	All	Dir.	Comp.	All	Dir.	Comp.	All	Dir.	Comp.	All	Dir.	Comp.	All
Proprietary Models															
Gemini-1.5-Flash	26.7	17.1	25.0	28.4	20.7	26.8	41.9	22.9	33.8	33.5	19.4	29.3	30.5	20.3	27.9
Gemini-1.5-Pro	42.1	21.4	38.5	28.4	20.7	26.8	41.9	22.9	33.8	33.5	19.4	29.3	36.0	21.2	32.2
Qwen-VL-Plus	22.7	11.4	20.8	28.8	13.8	25.5	33.3	9.4	23.1	27.2	13.4	23.1	27.0	11.9	23.1
Qwen-VL-Max	35.5	17.1	32.2	44.4	23.0	39.8	48.1	25.0	38.2	33.5	19.4	29.3	39.9	21.6	35.2
GPT-4-turbo	40.0	25.7	37.5	44.7	29.9	41.5	**55.0**	34.4	46.2	46.8	14.9	37.3	44.8	27.2	40.3
GPT-4o	**52.7**	**32.9**	**49.2**	**52.7**	**44.8**	**51.0**	53.5	**49.0**	**51.6**	**56.3**	**23.9**	**46.7**	**53.4**	**39.1**	**49.8**
Open-source Models															
Phi3-Vision-4B	43.3	27.1	40.5	47.9	27.6	43.5	33.3	27.1	30.7	50.0	14.9	39.6	44.6	24.7	39.5
QwenVL-Chat-7B	20.6	17.1	20.0	18.2	9.2	16.2	31.0	14.6	24.0	24.7	11.9	20.9	21.9	13.1	19.7
LlaVa1.6-Vicuna-7B	25.8	14.3	23.8	24.9	19.5	23.8	38.0	15.6	28.4	21.5	11.9	18.7	26.5	15.6	23.7
Intern-VL-2.0-8B	38.5	27.1	36.5	45.7	29.9	42.2	43.4	27.1	36.4	44.9	**22.4**	38.2	42.7	26.9	38.6
Llama3-Next-8B	20.3	5.7	17.8	22.4	16.1	21.0	24.0	21.9	23.1	20.9	16.4	19.6	21.6	15.6	20.1
CogVLM2-19B	24.8	11.4	22.5	28.8	10.3	24.8	27.9	5.2	18.2	24.7	7.5	19.6	26.6	8.4	21.9
LlaVa1.6-YI-34B	5.8	10.0	6.5	7.7	4.6	7.0	13.2	5.2	9.8	9.5	6.0	8.4	8.1	6.2	7.6
Intern-VL-2.0-40B	**53.3**	**42.9**	**51.5**	**54.3**	**37.9**	**50.7**	**55.8**	**37.5**	**48.0**	**51.9**	20.9	**42.7**	**53.8**	**35.3**	**49.0**
Chart Expert Models															
ChartLlama-13B	7.3	4.3	6.8	7.3	10.3	8.0	21.7	6.2	15.1	13.9	6.0	11.6	10.4	6.9	9.5
ChartAst-S-13B	12.4	12.9	12.5	14.4	14.9	14.5	14.7	7.3	11.6	13.9	12.0	12.0	13.7	10.6	12.9
ChartIns-Llama2-7B	17.9	11.4	16.8	16.0	19.5	16.8	27.1	13.5	21.3	13.9	9.0	12.4	17.8	13.8	16.8
ChartIns-FlanT5-3B	23.6	24.3	23.8	28.4	16.1	25.8	40.3	19.8	31.6	13.9	19.4	15.6	25.9	19.7	24.3
ChartGemma-2B	33.9	25.7	32.5	29.1	25.3	28.2	36.4	30.2	33.8	32.9	16.4	28.0	32.5	25.0	30.6
TinyChart-3B	24.5	15.7	23.0	28.4	17.2	26.0	33.3	15.6	25.8	33.5	17.9	28.9	28.6	16.6	25.5
EvoChart-4B	**62.1**	**32.9**	**57.0**	**62.3**	**33.3**	**56.0**	**64.3**	**30.2**	**49.8**	**55.1**	**37.3**	**49.8**	**61.3**	**33.1**	**54.2**

Table 2: Experimental results on EvoChart-QA using various open-source or proprietary models. Due to space constraints, abbreviations are used: Dir. refers to Direct, Comp. refers to Complex.

across all models at 54.2%. We summarize our findings as follows:

1) EvoChart-QA presents a substantially more challenging benchmark for evaluating basic chart comprehension. Even without involving numerical reasoning or calculation, these models experience a significant performance drop of 30-50% on EvoChart-QA compared to their scores on ChartQA. This challenge stems from the diversity of charts sourced from 140 websites and the meticulously crafted questions that comprise our dataset.

2) All models demonstrate significantly weaker performance on Complex Retrieval compared to Direct Retrieval, indicating that reasoning over visual information poses a substantially greater challenge than direct extraction of information from charts. Furthermore, nearly all models exhibit lower accuracy on Pie and Scatter chart types compared to their average performance. This suggests that Pie and Scatter charts pose a greater challenge.

3) Open-source general-purpose models exhibit comparable chart comprehension abilities to proprietary models. This suggests that models pretrained on large-scale chart data possess strong generalization capabilities. However, while chart expert models fine-tuned on specific domains achieve impressive scores on the ChartQA dataset, their performance degrades significantly to below 30% when confronted with the entirely OOD EvoChart-QA dataset.

EvoChart Results. Among all proprietary and open-source models evaluated, our proposed EvoChart trained model exhibits significantly superior performance, achieving an accuracy of 54.2%, surpassing GPT-4o 49.8%. We have the following observations:

Model	ChartQA	EvoChart-QA
Gemini-1.5-pro	81.3	32.2
GPT-4-turbo	62.3	40.3
GPT-4o	**85.7**	**49.8**
CogVLM2-19B	81.0	21.9
Phi3-Vision-4B	81.4	38.6
Intern-VL-2.0-8B	**81.5**	**49.0**
ChartAst-S-13B	79.9	12.9
TinyChart-3B	**83.6**	25.5
EvoChart-4B	81.5	**54.2**

Table 3: Comparison on ChartQA and EvoChart-QA

1) Although the EvoChart model is trained on synthetic data, it achieves SoTA performance on the entirely real-world benchmark EvoChart-QA and exhibits competitive performance on the chart reasoning task ChartQA. This validates the strong generalization ability of the EvoChart.

2) EvoChart primarily focuses on chart basic comprehension. However, as demonstrated in Table 3, EvoChart remains one of the top-performing chart expert models on

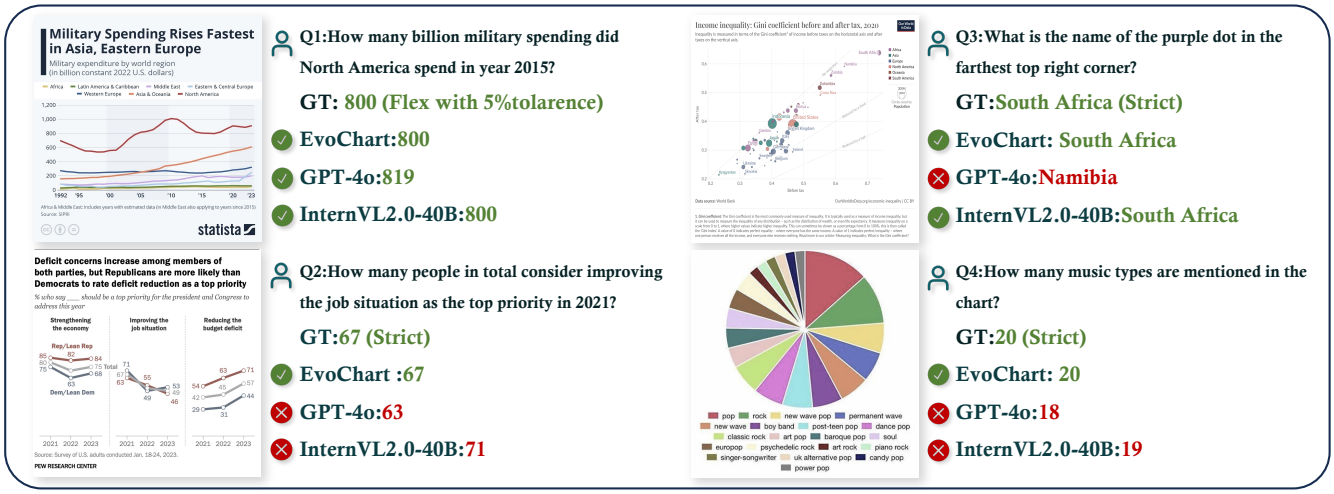

Figure 3: Four cases from the EvoChart-QA Benchmark. Q1 and Q2 are line charts, Q3 is a scatter chart, and Q4 is a pie chart.

ChartQA. This is an intriguing finding, suggesting that basic chart comprehension serves as a cornerstone for chart reasoning tasks, and training on basic comprehension can enhance performance in chart reasoning tasks.

3) EvoChart's complex retrieval capabilities are inferior to those of InternVL-2.0-40B and GPT4o. This is reasonable, as these models possess significantly larger scales, which confer an inherent advantage in complex visual extraction and reasoning tasks.

5.3 Analysis

EvoChart Ablation Study. We conducted comprehensive ablation studies on EvoChart, and the results are summarized in Table 4. We trained and generated EvoChart for 1 to 3 stages. Meanwhile, to verify the effectiveness of Chart Evaluation and Refinement, we set up EvoChart without refinement and trained it for 3 stages, denoted as "w/o refine stage-3." We observed the following:

1) As the number of EvoChart method Stages increases, the scale of the EvoChart-Dataset expands, and the performance of the EvoChart steadily improves. This highlights EvoChart's effectiveness as a self-training approach.

2) Despite having access to a larger training dataset, the "w/o refine stage-3." exhibits significantly lower performance on EvoChart-QA compared to the complete EvoChart method. This indicates the effectiveness of Chart Evaluation and Refinement in enhancing the model's generalization ability within the EvoChart.

Case Study. To further analyze EvoChart and EvoChart-QA, we selected samples from the EvoChart-QA Benchmark for analysis. Figure 3 presents four cases. More cases are provided in the Appendix. As shown in the figure, overall, EvoChart-QA offers diverse charts and questions, and EvoChart achieves more accurate chart understanding performance compared to GPT4o. Q2 and Q4 demonstrate the effectiveness of our Strict/Flex Metrics. For values explicitly labeled in the image, there should be zero tolerance. However, for estimation questions like Q1, a 5% tolerance is allowed. Furthermore, Q2 highlights EvoChart's capability for precise chart understanding in complex scenarios.

Model	Line	Bar	Pie	Scatter	Overall
w/ refine stage-1	53.2	51.5	46.7	44.9	50.0
w/ refine stage-2	53.5	54.2	**49.8**	48.0	52.0
w/ refine stage-3	**57.0**	**56.0**	**49.8**	**49.8**	**54.2**
w/o refine stage-3	52.5	50.7	43.6	45.3	49.0

Table 4: Ablation Study Results for EvoChart on EvoChart-QA

6 Conclusion

In this paper, we introduce EvoChart and EvoChart-QA: a novel approach for enhancing chart comprehension capabilities through self-training and iterative synthetic data generation, and a meticulously crafted real-world chart comprehension benchmark. We aim to provide a new avenue for real-world chart understanding through EvoChart and EvoChart-QA. Through extensive experimentation, we expose the limitations of existing VLMs in chart comprehension and validate the effectiveness of our EvoChart method across multiple datasets. In the future, we will further explore human-free methods in chart comprehension.

Acknowledgments

This work was supported by National Key Research and Development Program of China (2022YFC3303600), the Key Research and Development Project in Shaanxi Province No. 2022GXLH-01-03, National Natural Science Foundation of China (No. 62137002, 62293553, 62293554, 62450005, 62477036, 62293550, and 62306229), the Shaanxi Provincial Social Science Foundation Project (No. 2024P041), the Natural Science Basic Research Program of Shaanxi (No. 2023-JC-YB-593), the Youth Talent Support Program of Shaanxi Science and Technology Association (20240113), the China Postdoctoral Science Foundation (2024M752585).

References

Abdin, M.; Jacobs, S. A.; Awan, A. A.; Aneja, J.; and et al., A. A. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv:2404.14219.

Ahmed, S.; Jawade, B.; Pandey, S.; Setlur, S.; and Govindaraju, V. 2023. RealCQA: Scientific Chart Question Answering as a Test-Bed for First-Order Logic. In *ICDAR*, 14189: 66–83.

Baechler, G.; Sunkara, S.; Wang, M.; Zubach, F.; Mansoor, H.; Etter, V.; Carbune, V.; Lin, J.; Chen, J.; and Sharma, A. 2024. ScreenAI: A Vision-Language Model for UI and Infographics Understanding. In *IJCAI*, 3058–3068.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. arXiv:2308.12966.

Carbune, V.; Mansoor, H.; Liu, F.; Aralikatte, R.; Baechler, G.; Chen, J.; and Sharma, A. 2024. Chart-based Reasoning: Transferring Capabilities from LLMs to VLMs. In Duh, K.; Gómez-Adorno, H.; and Bethard, S., eds., *Findings of NAACL*, 989–1004.

Chen, Z.; Wu, J.; Wang, W.; Su, W.; Chen, G.; Xing, S.; Zhong, M.; Zhang, Q.; Zhu, X.; Lu, L.; Li, B.; Luo, P.; Lu, T.; Qiao, Y.; and Dai, J. 2023a. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. *arXiv preprint arXiv:2312.14238*.

Chen, Z.; Wu, J.; Wang, W.; Su, W.; Chen, G.; Xing, S.; Zhong, M.; Zhang, Q.; Zhu, X.; Lu, L.; Li, B.; Luo, P.; Lu, T.; and Qiao, Y. e. a. 2023b. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. *arXiv preprint arXiv:2312.14238*.

Cheng, Z.; Dai, Q.; and Hauptmann, A. G. 2023. ChartReader: A Unified Framework for Chart Derendering and Comprehension without Heuristic Rules. In *ICCV*, 22145–22156.

Dai, W.; Li, J.; Li, D.; Tiong, A. M. H.; Zhao, J.; Wang, W.; Li, B.; Fung, P.; and Hoi, S. C. H. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. In *NeurIPS*.

Gülçehre, Ç.; Paine, T. L.; Srinivasan, S.; Konyushkova, K.; Weerts, L.; Sharma, A.; Siddhant, A.; Ahern, A.; Wang, M.; Gu, C.; Macherey, W.; Doucet, A.; Firat, O.; and de Freitas, N. 2023. Reinforced Self-Training (ReST) for Language Modeling. *arXiv preprint arXiv:2308.08998*.

Han, Y.; Zhang, C.; Chen, X.; Yang, X.; Wang, Z.; Yu, G.; Fu, B.; and Zhang, H. 2023. ChartLlama: A Multimodal LLM for Chart Understanding and Generation. *arXiv preprint arXiv:2311.16483*.

Kafle, K.; Price, B. L.; Cohen, S.; and Kanan, C. 2018. DVQA: Understanding Data Visualizations via Question Answering. In *CVPR*, 5648–5656.

Kafle, K.; Shrestha, R.; Price, B. L.; Cohen, S.; and Kanan, C. 2020. Answering Questions about Data Visualizations using Efficient Bimodal Fusion. In *WACV*, 1487–1496.

Kahou, S. E.; Michalski, V.; Atkinson, A.; Kádár, Á.; Trischler, A.; and Bengio, Y. 2018. FigureQA: An Annotated Figure Dataset for Visual Reasoning. In *ICLR*.

Li, D.; Mei, H.; Shen, Y.; Su, S.; Zhang, W.; Wang, J.; Zu, M.; and Chen, W. 2018. ECharts: A declarative framework for rapid construction of web-based visualization. *VI*, 2(2): 136–146.

Li, F.; Zhang, R.; Zhang, H.; Zhang, Y.; Li, B.; Li, W.; Ma, Z.; and Li, C. 2024. LLaVA-NeXT-Interleave: Tackling Multi-image, Video, and 3D in Large Multimodal Models. *arXiv preprint arXiv:2407.07895*.

Lin, Z.; Liu, C.; Zhang, R.; Gao, P.; Qiu, L.; Xiao, H.; Qiu, H.; Lin, C.; Shao, W.; Chen, K.; Han, J.; Huang, S.; Zhang, Y.; He, X.; Li, H.; and Qiao, Y. 2023. SPHINX: The Joint Mixing of Weights, Tasks, and Visual Embeddings for Multi-modal Large Language Models. *arXiv preprint arXiv:2311.07575*.

Liu, F.; Piccinno, F.; Krichene, S.; Pang, C.; Lee, K.; Joshi, M.; Altun, Y.; Collier, N.; and Eisenschlos, J. M. 2023a. MatCha: Enhancing Visual Language Pretraining with Math Reasoning and Chart Derendering. In *ACL*, 12756–12770.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024. Improved baselines with visual instruction tuning. In *CVPR*, 26296–26306.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023b. Visual Instruction Tuning. In *NeurIPS*.

Masry, A.; Kavehzadeh, P.; Long, D. X.; Hoque, E.; and Joty, S. 2023. UniChart: A Universal Vision-language Pretrained Model for Chart Comprehension and Reasoning. In *EMNLP*, 14662–14684.

Masry, A.; Long, D. X.; Tan, J. Q.; Joty, S. R.; and Hoque, E. 2022. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In *Findings of ACL*, 2263–2279.

Masry, A.; Shahmohammadi, M.; Parvez, M. R.; Hoque, E.; and Joty, S. 2024a. ChartInstruct: Instruction Tuning for Chart Comprehension and Reasoning. *arXiv preprint arXiv:2403.09028*.

Masry, A.; Thakkar, M.; Bajaj, A.; Kartha, A.; Hoque, E.; and Joty, S. 2024b. ChartGemma: Visual Instruction-tuning for Chart Reasoning in the Wild. arXiv:2407.04172.

Meng, F.; Shao, W.; Lu, Q.; Gao, P.; Zhang, K.; Qiao, Y.; and Luo, P. 2024. ChartAssisstant: A Universal Chart Multimodal Language Model via Chart-to-Table Pre-training and Multitask Instruction Tuning. *arXiv preprint arXiv: 2401.02384*.

Methani, N.; Ganguly, P.; Khapra, M. M.; and Kumar, P. 2020. PlotQA: Reasoning over Scientific Plots. In *WACV*, 1516–1525.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; and et al., F. L. A. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Rose, D.; Himakunthala, V.; Ouyang, A.; He, R.; Mei, A.; Lu, Y.; Saxon, M.; Sonar, C.; Mirza, D.; and Wang, W. Y. 2024. Visual Chain of Thought: Bridging Logical Gaps with Multimodal Infillings. arXiv:2305.02317.

Team, G.; Georgiev, P.; Lei, V. I.; Burnell, R.; Bai, L.; Gulati, A.; Tanzer, G.; Vincent, D.; Pan, Z.; Wang, S.; and et al., S. M. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv:2403.05530.

Ulmer, D.; Mansimov, E.; Lin, K.; Sun, J.; Gao, X.; and Zhang, Y. 2024. Bootstrapping LLM-based Task-Oriented Dialogue Agents via Self-Talk. arXiv:2401.05033.

Wang, W.; Lv, Q.; Yu, W.; Hong, W.; Qi, J.; Wang, Y.; Ji, J.; Yang, Z.; Zhao, L.; Song, X.; Xu, J.; Xu, B.; Li, J.; Dong, Y.; Ding, M.; and Tang, J. 2023. CogVLM: Visual Expert for Pretrained Language Models. arXiv:2311.03079.

Wang, Z.; Xia, M.; He, L.; Chen, H.; Liu, Y.; Zhu, R.; Liang, K.; Wu, X.; Liu, H.; Malladi, S.; Chevalier, A.; Arora, S.; and Chen, D. 2024. CharXiv: Charting Gaps in Realistic Chart Understanding in Multimodal LLMs. *arXiv preprint arXiv:2406.18521*.

Xia, R.; Zhang, B.; Ye, H.; Yan, X.; Liu, Q.; Zhou, H.; Chen, Z.; Dou, M.; Shi, B.; Yan, J.; and Qiao, Y. 2024. ChartX & ChartVLM: A Versatile Benchmark and Foundation Model for Complicated Chart Reasoning. *arXiv preprint arXiv:2402.12185*.

Xu, F.; Sun, Q.; Cheng, K.; Liu, J.; Qiao, Y.; and Wu, Z. 2024. Interactive Evolution: A Neural-Symbolic Self-Training Framework For Large Language Models. *arXiv preprint arXiv:2406.11736*.

Xu, F.; Wu, Z.; Sun, Q.; Ren, S.; Yuan, F.; Yuan, S.; Lin, Q.; Qiao, Y.; and Liu, J. 2023. Symbol-LLM: Towards Foundational Symbol-centric Interface For Large Language Models. *arXiv preprint arXiv:2311.09278*.

Yoo, K. M.; Park, D.; Kang, J.; Lee, S.; and Park, W. 2021. GPT3Mix: Leveraging Large-scale Language Models for Text Augmentation. In *Findings of EMNLP*, 2225–2239.

Zhang, L.; Hu, A.; Xu, H.; Yan, M.; Xu, Y.; Jin, Q.; Zhang, J.; and Huang, F. 2024. TinyChart: Efficient Chart Understanding with Visual Token Merging and Program-of-Thoughts Learning. *arXiv preprint arXiv: 2404.16635*.

Zhu, W.; Agarwal, A.; Joshi, M.; Jia, R.; Thomason, J.; and Toutanova, K. 2024. Efficient End-to-End Visual Document Understanding with Rationale Distillation. In Duh, K.; Gómez-Adorno, H.; and Bethard, S., eds., *In NAACL*, 8401–8424.