Chart-R1: Chain-of-Thought Supervision and Reinforcement for Advanced Chart Reasoner

Lei Chen[1] **Xuanle Zhao**[2] **Zhixiong Zeng**[1†] **Jing Huang**[1] **Yufeng Zhong**[1] **Lin Ma**[1*]
[1] Meituan [2] Institute of Automation, Chinese Academy of Sciences
leichen1997@outlook.com zengzhixiong@meituan.com forest.linma@gmail.com

Abstract

Recently, inspired by OpenAI-o1/o3 and Deepseek-R1, the R1-Style method based on reinforcement learning fine-tuning has received widespread attention from the community. Previous R1-Style methods mainly focus on mathematical reasoning and code intelligence. It is of great research significance to verify their advantages on more general multimodal data. Chart is an important multimodal data type with rich information, which brings important research challenges in complex reasoning. In this work, we introduce Chart-R1, a chart-domain vision-language model with reinforcement learning fine-tuning to enable complex chart reasoning. To support Chart-R1, we first propose a novel programmatic data synthesis technology to generate high-quality step-by-step chart reasoning data covering single- and multi-subcharts, which makes up for the lack of reasoning data in the chart domain. Then we develop a two-stage training strategy: Chart-COT with step-by-step chain-of-thought supervision, and Chart-RFT with numerically sensitive reinforcement fine-tuning. Chart-COT aims to decompose complex chart reasoning tasks into fine-grained, understandable subtasks through step-by-step supervision, which lays a good foundation for improving the reasoning level of reinforcement learning. Chart-RFT utilize the typical group relative policy optimization strategy, in which a relatively soft reward is adopted for numerical response to emphasize the numerical sensitivity in the chart domain. We conduct extensive experiments on open-source benchmarks and self-built chart reasoning dataset (*i.e., ChartRQA*). Experimental results show that Chart-R1 has significant advantages compared to chart-domain methods, even comparable to open/closed source large-scale models (*e.g., GPT-4o, Claude-3.5*). Our code and dataset will be available at `https://github.com/DocTron-hub/Chart-R1`.

1 Introduction

Recently, inspired by the success of models such as OpenAI's o1/o3 OpenAI [2025] and DeepSeek-R1 Guo et al. [2025], leveraging Reinforcement Learning (RL) for fine-tuning has garnered significant attention within the research community. Although these methods have shown promise in textual domains like mathematical reasoning, code generation, and multidisciplinary knowledge, transferring these advanced reasoning capabilities to the vision domain presents an open challenge. While recent approaches like Vision-R1 Huang et al. [2025] and VLM-R1 Shen et al. [2025] have successfully leveraged RL to enhance visual perception and grounding, they have primarily focused on simple questions, neglecting tasks that demand deep reasoning capabilities.

Charts, as information-intensive images, are a crucial research area in image understanding and reasoning Wang et al. [2024]. Prior works improve chart perception and understanding capacities by

† Project leader.
* Corresponding author.

Preprint.

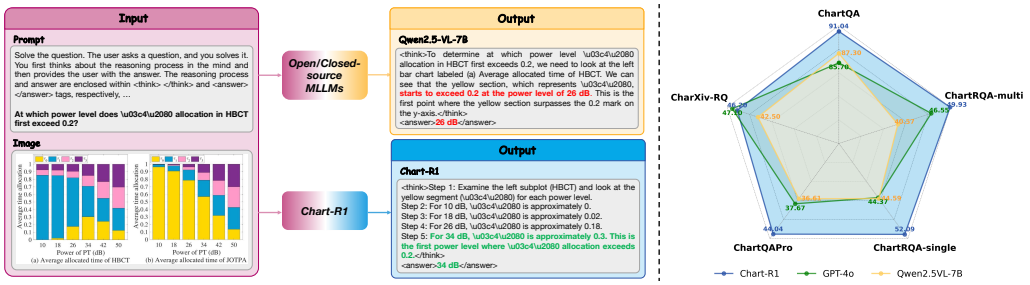


Figure 1: Comparison of existing VLM performance on chart understanding and reasoning benchmarks. In the complex chart reasoning task, Qwen2.5-VL-7B generates a wrong thinking process, whereas Chart-R1 thinks and answers correctly.

supervised fine-tuning (SFT) on datasets augmented with Chain-of-Thought (CoT) or Program-of-Thought (PoT) methods Wei et al. [2022], Chen et al. [2022]. A key limitation of SFT is that it causes models to overfit specific reasoning patterns, hindering their generalization abilities. Following the DeepSeek R1, recent methods Jia et al. [2025], Ni et al. [2025] leverage RL to enhance VLM reasoning capabilities. However, the scope of these efforts has been largely limited to visual perception and understanding, rather than the complex reasoning required for deep chart analysis.

In this work, we propose Chart-R1, a chart domain VLM that leverages RL to enhance complex reasoning capability. To this end, we introduce two key contributions. First, we propose a novel programmatic synthesis strategy to generate high-quality reasoning data. Second, we introduce an effective two-stage training strategy that significantly enhances reasoning capacity. Specifically, to support Chart-R1 training, we first generate complex chart reasoning data in the programmatic synthesis method. We utilize LLMs to generate the chart plotting code and then use the generated code to formulate complex questions, multi-step CoT reasoning processes, and the final answer. To this end, we construct ChartRQA, a complex reasoning dataset featuring 258k multi-step reasoning samples that cover both single- and multi-chart tasks. To ensure the fidelity of the data in charts, we curate real-world tables from arXiv papers as the data source. ChartReasoner Jia et al. [2025] proposes converting charts into code to generate reasoning data. However, its reliance on a lossy parsing process directly restricts the scope and diversity of the final reasoning data. The training of Chart-R1 is conducted in two stages: Chart-COT with step-by-step chain-of-thought supervision, and Chart-RFT with numerically sensitive reinforcement fine-tuning. During the initial Chart-COT stage, the model is fine-tuned via SFT on step-by-step reasoning data to build its core capability of decomposing complex tasks into fine-grained subtasks. In the Chart-RFT stage, we utilize the group relative policy optimization (GRPO) strategy, where the reward signal is a composite of soft matching and edit distance. This design specifically targets and enhances accuracy for both numerical and string-based answers. Notably, distinct datasets are employed for these two stages, based on our finding that training on the same data impairs the model's exploration ability during the RL process. Furthermore, we introduce a human-verified benchmark, ChartRQA, to assess the boundaries of complex chart reasoning. Unlike prior works Xia et al. [2024], Wang et al. [2024], its questions feature a higher degree of complexity, requiring multi-step thought processes. The substantial performance drop of existing VLMs on ChartRQA exposes a critical limitation in their chart reasoning capabilities.

In summary, our contributions are as follows:

- We propose a novel two-stage training strategy, consisting of Chart-COT and Chart-RFT, to enhance chart reasoning in VLMs. Our model, Chart-R1, trained with this strategy, establishes a new SOTA on various chart understanding and reasoning benchmarks.
- We introduce a programmatic data synthesis strategy that leverages code as a pivotal starting source to generate step-by-step reasoning data. The data source is grounded in real-world tables from arXiv papers, ensuring high fidelity in the resulting charts.
- We introduce ChartRQA, a comprehensive dataset for complex chart reasoning that includes a human-verified benchmark and a large-scale training dataset. The substantial performance of existing VLMs on the ChartRQA benchmark underscores a critical limitation in their chart reasoning capabilities.

- We conduct a series of comprehensive experiments to systematically assess the impact of various settings. Our findings provide valuable insights and offer clear guidance for future research in this domain.

2 Related Works

2.1 Chart VLMs

Chart understanding and reasoning are crucial areas of research community that encompass both low-level and high-level tasks Singh et al. [2019], Methani et al. [2020]. Recently, many chart-domain models have been proposed to enhance the chart understanding capacity of VLMs Han et al. [2023], Liu et al. [2023]. However, prior works have concentrated on descriptive tasks Masry et al. [2024a,b], such as extracting explicit content from charts Masry et al. [2022]. In contrast, more recent works focus on leveraging the reasoning capabilities of VLMs to interpret complex and implicit information within the charts. For example, TinyChart Zhang et al. [2024] utilizes a template-based method to generate the Program-of-Thought (PoT) Chen et al. [2022] reasoning data. ChartCoder Zhao et al. [2025b] proposes Snippet-of-Thought to enhance chart-to-code generation. ChartReasoner Jia et al. [2025] utilizes a chart-to-code model to convert chart images into code and generate the reasoning process based on code. However, the generated reasoning data has limitations due to the chart-to-code accuracy Shi et al. [2024], Xu et al. [2024].

2.2 Long Reasoning VLMs

Recently, with the success of DeepSeek-R1 Guo et al. [2025], many works have attempted to enhance the LLM reasoning ability via rule-based reward and RL Shao et al. [2024]. In the vision-language domain, recent works follow the DeepSeek-R1 method to enhance the long-chain reasoning capacity of VLMs Shen et al. [2025], Wang et al. [2025], Qiu et al. [2025]. For example, Vision-R1 Huang et al. [2025] and R1-OneVision Yang et al. [2025] apply Group Relative Policy Optimization (GRPO) with multimodal reasoning data to enable VLMs for long reasoning. MMEureka Meng et al. [2025b] and R1-Zero Liu et al. [2025] further advance the visual long-term reasoning with improved RL training strategies. Point-RFT Ni et al. [2025] utilizes grounded CoT reasoning for visual understanding, but it just utilize ChartQA for RL which limits the final model reasoning capacity.

2.3 Chart Understanding and Reasoning

A variety of training datasets and evaluation benchmarks have been developed to improve VLM performance on chart-related tasks Xia et al. [2024], Shi et al. [2024], He et al. [2024], Zhao et al. [2025a], Wu et al. [2025]. Previous works generally focus on description tasks, for example, ChartQA Masry et al. [2022], PlotQA Methani et al. [2020] and Chart-to-text Kantharaj et al. [2022] mainly train and evaluate the capacities of the models on extracting information from the chart. While numerous relevant works exist, the challenge in the description tasks is predominantly driven by chart complexity. Recent works such as Charxiv Wang et al. [2024] and CharMuseum Tang et al. [2025] introduce more challenging reasoning tasks, demanding that models think before answering. Unlike descriptive tasks, reasoning tasks present a dual challenge, originating from both the perceptual complexity of the chart and the reasoning depth required by questions.

3 Method

To enhance the reasoning capabilities of models on chart reasoning tasks, we introduce our proposed data synthesis and two-stage training strategy. We first programmatically generate a large-scale training dataset with the CoT reasoning process and subsequently employ the SFT on CoT data as a cold start phase to bootstrap the subsequent RL strategy for training.

3.1 Programatic Data Synthesis

While several CoT datasets for chart reasoning have been proposed, they are largely derivatives of the ChartQA dataset, constructed by augmenting its existing question-answer pairs with generated reasoning processes Zhang et al. [2024], Jia et al. [2025]. However, this method is akin to distilling

Table 1: Comparison of our proposed ChartRQA training set with other chart datasets. ChartRQA features the integration of single/multi-charts, thinking processes, and verifiable answer formats.

Dataset	Types	Unique Charts	Multi-chart	Thinking Process
ChartQA Masry et al. [2022]	3	21.9k	✗	✗
MMC Liu et al. [2023]	7	600k	✔	✗
ChartLlama Han et al. [2023]	10	11k	✗	✗
NovaChart Hu et al. [2024]	18	47k	✗	✔
ChartRQA (Ours)	24	93.3k	✔	✔

reasoning from SOTA VLMs, which is problematic as the failures of these models on complex tasks inherently limit the quality of the generated data. Generating high-quality CoT reasoning data is a well-recognized challenge, largely because current methods use the final answer's correctness as the sole supervisory signal. This problem is particularly acute in the domain of complex chart reasoning, as existing models already exhibit significant limitations. Consequently, data generated via this approach inherently suffers from both low quality and limited diversity. Although the recent ChartReasoner method Jia et al. [2025] generates reasoning data by first parsing charts into code, the diversity and quality of generated data is fundamentally limited by the performance of the chart-to-code parser. In contrast, our programmatic data generation strategy reverses this paradigm by utilizing code as a pivotal starting source. First, we prompt a powerful LLMs to generate plotting code. This code then serves as a perfect, high-fidelity foundation from which the LLM subsequently synthesizes question-answer pairs and their complex step-by-step reasoning path.

Plotting Code Generation We instruct LLMs to generate Matplotlib plotting code to render high-quality and diverse chart images. However, our analysis reveals that directly generating synthetic data values in plotting code often yields monotonous trends that lack complexity and diversity. To address this, we first curate tables from real-world arXiv papers, which serve as veritable data sources. Secondly, to enhance the diversity of the generated code, we manually write seed code examples for different chart types. To ensure the diversity of generated code, we randomly combine the curated table and seed code as in-context learning sources for LLMs to generate plotting code. To generate complex, multi-chart scenarios, we both include numerous multi-chart examples in our seed code and explicitly prompt the LLM during generation to use functions like plt.subplots() to create composite figures. Our work significantly expands the range of chart types available for chart reasoning, representing the most diverse dataset. We execute all generated code samples and discard any that fail to run successfully.

Reasoning Data Generation With the executable plotting code as a foundation, we prompt LLMs to synthesize a complete reasoning instance, comprising a question, its answer, and a step-by-step reasoning path. To increase the diversity, we separate the plotting code into single- and multi-chart ones and utilize distinct instructions for instance generation. For multi-chart problems in particular, we prompt the LLM to generate questions requiring information to be cross-referenced between sub-charts. The results show that this strategy significantly enhances multi-chart task complexity. Our results show that leveraging code allows LLMs to produce more complex questions and detailed reasoning compared to methods that use chart images alone. We posit that a code-based approach is superior for generating complex chart reasoning as the underlying code provides a lossless textual representation of details while enabling the scalable synthesis of new data independent of existing corpora. We filter out data samples that do not conform to the thinking and answering formats and faulty chart images.

Dataset Construction Using the aforementioned methods, we construct ChartRQA, a comprehensive chart reasoning corpus that includes a large-scale training dataset of 258k instances with reasoning paths as well as a human-verified benchmark. The training dataset is separated into two subsets for our two-stage training strategy, ChartRQA-SFT and ChartRQA-RL, consisting of 228k and 30k samples, respectively. Detailed comparisons about ChartRQA with other chart-domain training set are denoted in Table 1. The benchmark is constructed via a human validation where experts review each sample for question difficulty and answer correctness, subsequently constructing 1,702 high-quality samples (933 single-chart and 769 multi-chart tasks) for evaluation. As detailed in Table 2, we also calculated the average token counts for the questions, reasoning paths, and final answers, broken down by single- and multi-chart problems. The analysis reveals that the components associated with multi-chart problems are significantly longer than those for single-chart problems. Also, the

Table 2: The average question, thinking process, and answer lengths in the ChartRQA train and test sets. We count the single- and multi-chart problems of each set separately.

Token Avg.	**Train**			**Test**		
	Single	Multi	Total	Single	Multi	Total
Question	30.03	39.84	34.03	29.83	39.49	34.19
Thinking Process	196.50	237.38	213.17	196.32	240.94	216.48
Answer	5.98	8.87	7.16	5.96	8.97	7.32

distribution between the train and test sets is balanced. Figure 2 is the showcase of our generated ChartRQA.

Quality Evaluation To assess the quality of our generated data, we randomly sample 1k instances and recruit human experts for evaluation. The results indicate that over 85% of the instances are free from errors. Notably, we deliberately omit any data cleaning process. The fact that our model, Chart-R1, achieves strong performance despite being trained on this raw, uncurated dataset validates the robustness of our proposed code-based generation strategy.

3.2 Chart-COT

To enhance the chart reasoning capacity, we propose a two-stage training strategy. Utilizing Qwen2.5VL-7B-Instruct as the baseline model, we first SFT it on the step-by-step reasoning data of our proposed ChartRQA-SFT. Specifically, the baseline model first undergoes SFT on our generated step-by-step reasoning data, which serves as the code-starting phase to equip the model with the fundamental capability to decompose complex tasks into fine-grained subtasks. Our ablation studies demonstrate that a preliminary SFT stage on CoT data is critical, as it yields significantly better performance than applying RL from scratch.

We train the model using a standard autoregressive language modelling objective. The loss function is the negative log-likelihood of the target sequence:

$$\mathcal{L}(\theta) := -\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{CoT}}} \sum_{t=1}^{T} \log P\left(y_t \mid x, y_{<t}; \theta\right), \tag{1}$$

where (x, y) is the query and target response, with the reasoning process.

3.3 Chart-RFT

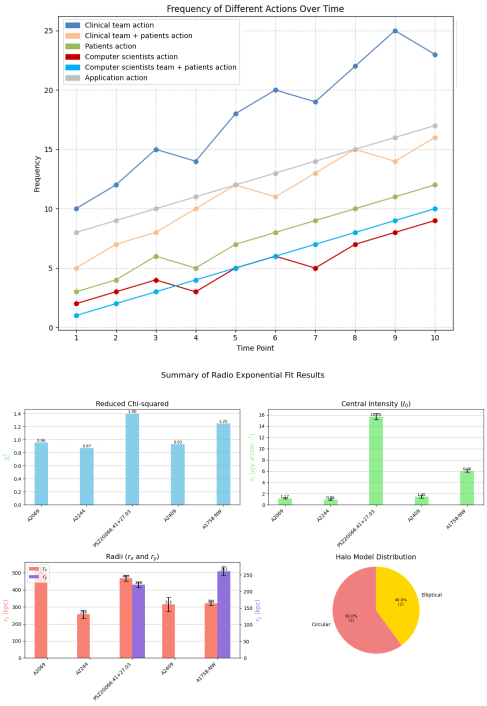
After the Chart-COT stage, while the fine-tuned model demonstrates an enhanced ability to decompose complex questions, its performance on out-of-domain (OOD) tasks notably degrades. We hypothesize this is due to a distributional mismatch between ChartRQA-SFT with some simple chart understanding tasks, which harms its generalization ability. To address the degradation in generalization, we subsequently apply reinforcement fine-tuning (RFT) to generalize its reasoning capacity.

Group Relative Policy Optimization Following recent reasoning works Guo et al. [2025], we adapt the Group Relative Policy Optimization (GRPO) Shao et al. [2024] algorithm for RFT. GRPO foregoes the critic model, instead estimating the baseline from group scores, significantly reducing training resources. For each input (x, y), the policy π_θ samples a group of G candidate responses $\{o_i\}_{i=1}^{G}$.

$$\begin{aligned}\mathcal{J}_{GRPO}(\theta) =& \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{CoT}},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(O|x)} \Bigg[\frac{1}{G}\sum_{i=1}^{G}\min\Bigg(\frac{\pi_\theta(o_i \mid x)}{\pi_{\theta_{\text{old}}}(o_i \mid x)}A_i, \\ & \text{clip}\left(\frac{\pi_\theta(o_i \mid x)}{\pi_{\theta_{\text{old}}}(o_i \mid x)}, 1-\varepsilon, 1+\varepsilon\right)A_i\Bigg) - \beta\mathbb{D}_{KL}(\pi_\theta\|\pi_{\text{SFT}})\Bigg]\end{aligned} \tag{2}$$

where ε and β are hyperparameters, and π_{SFT}, π_θ, and $\pi_{\theta_{\text{old}}}$ are the model after SFT, the optimized model and the old policy model. The group-normalized advantage for the i-th response is:

$$A_i = \frac{r_i - \text{mean}\left(\{r_1, r_2, \cdots, r_G\}\right)}{\text{std}\left(\{r_1, r_2, \cdots, r_G\}\right)} \tag{3}$$



Question: Which action shows the largest percentage increase in frequency from Time Point 1 to Time Point 10 among 'Patients action' and 'Computer scientists team + patients action'?

Answer:
<think>
Step 1: For 'Patients action' (green line), find the frequency at Time Point 1, which is 3.
Step 2: Find the frequency for 'Patients action' at Time Point 10, which is 12.
Step 3: Calculate the percentage increase for 'Patients action': ((12 - 3) / 3) * 100 = (9 / 3) * 100 = 300%.
Step 4: For 'Computer scientists team + patients action' (light blue line), find the frequency at Time Point 1, which is 1.
Step 5: Find the frequency for 'Computer scientists team + patients action' at Time Point 10, which is 10.
Step 6: Calculate the percentage increase for 'Computer scientists team + patients action': ((10 - 1) / 1) * 100 = (9 / 1) * 100 = 900%.
Step 7: Compare the percentage increases: 900% is larger than 300%.
</think>
<answer>Computer scientists team + patients action</answer>

Question: For the object with the highest Central Intensity (I_0), what is its r_x radius, and what percentage of total halo models are circular?

Answer:
<think>
Step 1: Identify the object with the highest Central Intensity (I_0) from the 'Central Intensity (I_0)' bar chart. The values are 1.17, 0.96, 15.70, 1.45, 6.06. The highest I_0 is 15.70, corresponding to object PSZ2G066.41+27.03.
Step 2: Find the r_x radius for PSZ2G066.41+27.03 from the 'Radii (r_x and r_y)' bar chart. The r_x value for PSZ2G066.41+27.03 is 468 kpc.
Step 3: Determine the percentage of total halo models that are circular from the 'Halo Model Distribution' pie chart. The pie chart shows 'Circular' as 60.0%.
</think>
<answer>468 kpc and 60.0%</answer>

Figure 2: Showcases of our proposed ChartRQA dataset. The ChartRQA include single- and multi-chart images with complex questions that need step-by-step thinking processes to answer.

Reward Design For effective RFT, we follow the DeepSeek-R1 Shao et al. [2024] and adopt a rule-based reward that consists of accuracy and format rewards. We introduce a soft accuracy reward for chart problems, which utilizes distinct functions to evaluate numerical and string-based tasks separately. This allows for a more appropriate assessment based on the expected answer type.

- **Accuracy Reward.** We employ distinct reward functions to measure the correctness of model outputs, with each function tailored to the specific answer type. In the case of numerical answers, we adopt the soft matching technique from Point-RFT Ni et al. [2025], allowing for a relative error tolerance of $\pm 5\%$. For string-based answers, we utilize the edit distance as the reward signal.
- **Format Reward.** The format reward is determined using a grammar-level regex parser. This parser validates the structural integrity of the output by checking two conditions: (1) that the reasoning process is correctly enclosed within <think> and </think> tags, and (2) that the final answer can be extracted from the designated answer tag <answer> and </answer>.

Data Proportion For the Chart-COT and Chart-RFT stages, we utilize distinct subsets of ChartRQA. This setting is critical, as our experiments reveal that using the same CoT data for both phases causes the model to overfit to replicate the reasoning paths from the SFT data, which in turn degrades the diversity and exploration capability of the policy model during the RL phase. We find that the stability and convergence of the Chart-RFT phase critically depend on the pattern consistency of the data from the preceding Chart-COT stage. Employing SFT data with inconsistent patterns significantly hinders RFT convergence, highlighting the necessity of a distributionally aligned dataset in the Chart-COT stage to ensure effective downstream RFT.

4 Experiments

4.1 Implementation Details

For data generation, we employ Gemini-2.5-Flash to create both plotting code and QA pairs. In the training stage, our ChartRQA-SFT is used for SFT, while a combination of ChartQA and ChartRQA-RL is used for GRPO. The SFT stage is trained for one epoch with a batch size of 48, and the RL

Table 3: The main results on existing chart understanding and reasoning benchmarks. Our proposed Chart-R1 outperforms all the small-scale VLMs (<20B) on the evaluation benchmarks. **Bold** denotes the best performances of open-source VLMs.

Model Name	ChartQA	CharXiv-RQ	ChartQAPro	ChartRQA (single / multi)
Proprietary				
GPT-4o	85.7	47.1	37.67	44.37 / 46.55
Gemini-1.5-Flash	79.0	33.9	42.96	-
Gemini-1.5-Pro	87.2	43.3	-	-
Gemini-2.5-Flash	-	-	-	59.12 / 59.17
Claude-3.5-Sonnet	90.8	60.2	43.58	52.79 / 56.05
General-domain Open-source				
Phi-3.5-Vision	81.8	32.7	24.73	31.08 / 24.32
DeepSeek-VL2	86.0	-	16.28	23.15 / 20.29
InternVL3-8B	86.6	37.6	-	37.51 / 31.73
InternVL3-38B	89.2	46.4	-	46.09 / 38.36
Qwen2.5-VL-7B	87.3	42.5	36.61	44.59 / 40.57
Chart-domain				
ChartLlama	69.66	14.2	-	-
TinyChart	83.60	8.3	13.25	6.75 / 6.11
ChartGemma	80.16	12.5	6.84	7.18 / 9.23
ChartReasoner	86.93	-	39.97	-
Chart-R1-7B (Ours)	**91.04**	**46.2**	**44.04**	**52.09 / 49.93**

stage is trained for 3 epochs with a batch size of 128. The final Chart-R1 model is obtained by applying this RL process to the initial SFT-trained model, Chart-R1-SFT. For these respective stages, the learning rates are set to 1e-5 and 1e-6. Finally, the training processes for SFT and RL required approximately 3 and 30 hours, respectively, on a system with 24 H800 GPUs.

4.2 Experiment Settings

We conduct experiments to evaluate the results obtained from various training settings. Firstly, we assess the training stages and the scope of training data, including: (1) SFT with CoT data, (2) Directly RL versus CoT-RL, and (3) RL w/ and w/o the ChartRQA data.

Benchmarks To comprehensively evaluate the understanding and reasoning capacity of our posed Chart-R1, we choose ChartQA Masry et al. [2022], Chaxiv-RQ (Reasoning Questions) Wang et al. [2024], ChartQAPro Masry et al. [2025] and our proposed ChartRQA as the evaluation benchmarks.

Baselines We compare our proposed Chart-R1 with existing models in three setups: (1) Proprietary models include GPT-4oOpenAI [2024], Gemini-1.5-(Flash, Pro)Team et al. [2023], Gemini-2.5-Flash and Claude-3.5-SonnetAnthropic [2024]. (2) General-domain open-source VLMs including Phi 3.5-Vision Abdin et al. [2024], DeepSeek-VL2 Wu et al. [2024], InternVL3(8B, 38B) Zhu et al. [2025] and Qwen2.5-VL(7B) Bai et al. [2025]. (3) Chart-domain VLMs including ChartLlama Han et al. [2023], TinyChart Zhang et al. [2024], ChartGemma Masry et al. [2024b] and ChartResoner Jia et al. [2025].

4.3 Main Results

Table 3 show the performance of Chart-R1 compared with other baseline models. The results show that Chart-R1 achieve the state-of-the-art performance on small-scale (<20B) VLMs, including general- and chart-domain models across all the benchmarks. Especially in ChartQA, Chart-R1 achieves the best performance, even compared with proprietary and large-scale VLMs. In the chart reasoning benchmark, CharXiv-RQ, ChartQAPro and our proposed ChartRQA, Chart-R1 significantly surpass existing chart-domain models. Since the training data of Chart-R1 only contains ChartRQA and ChartQA, these results demonstrate the diversity of our proposed ChartRQA dataset and CoT-RL training strategy.

Table 4: The ablation study about different SFT and RL training settings. QA and RQA are the abbreviations of ChartQA and ChartRQA.

Model Name	Training Setting SFT	Training Setting RL	ChartQA	CharXiv-RQ	ChartRQA (single / multi)
Qwen2.5-VL-7B			87.3	42.5	44.59 / 40.57
Qwen2.5-VL-7B-SFT	*QA*		86.16	36.0	24.76 / 18.34
Qwen2.5-VL-7B-RL		*QA*	89.32	42.1	37.73 / 36.15
		QA+RQA-RL	90.28	45.2	44.16 / 40.44
	RQA-SFT	*QA+RQA-RL*	**91.04**	**46.2**	**52.09 / 49.93**

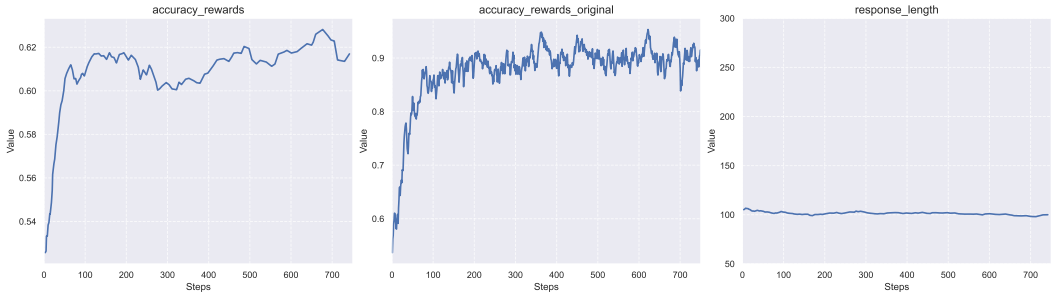


Figure 3: The training curve during the RL stage that utilizing the ChartQA dataset solely.

4.4 Ablation Study

We first assess the impact of different training settings, with results presented in Table 4. The findings indicate that utilizing our two-stage training strategy, Chart-SFT data for step-by-step SFT and ChartQA and ChartRQA-RL data for RL yields the most balanced performance. Notably, omitting Chart-COT causes a significant performance drop on the ChartRQA benchmark. We attribute this to the nature of ChartRQA as complex charts requiring multi-step thinking before answering. The first Chart-COT stage equips the model with the necessary capability for such step-by-step task decomposition. Also, SFT exclusively on the ChartQA dataset leads to performance degradation across all benchmarks, including ChartQA itself. We reckon that although SFT could improve capacity for in-domain tasks, training on simple and low-diversity datasets disrupts the tuned distribution, harming the ability on both in-domain (ChartQA) and OOD (CharXiv-RQ, ChartRQA) tasks.

Previous research has established that the complexity of the training data is critical for effective RL Guo et al. [2025]. Our generated ChartRQA training set meets this requirement, featuring tasks with single- and multi-chart images and questions requiring step-by-step reasoning. Including our ChartRQA dataset during the RL stage is crucial for achieving optimal performance. The structural and logical complexity is important for performance enhancements are observed in our Chart-RFT stage. We find that training exclusively on the ChartQA dataset is insufficient for developing a robust reasoning model. The limited complexity of ChartQA fails to encourage the model to learn diverse, long-path reasoning strategies. This limitation is empirically demonstrated by the training process shown in Figure 3. The accuracy reward rapidly converges to around 0.9 with little subsequent growth, while the response length remains constrained to approximately 100 tokens.

Table 5: Ablation study on the accuracy reward. The results show that utilizing specialized reward functions for different task types achieves superior performance.

Accuracy Reward Edit Distance	Accuracy Reward Soft Matching	ChartQA	CharXiv-RQ	ChartRQA (single / multi)
✓		89.88	44.0	**45.02** / 39.79
✓	✓	**90.28**	**45.2**	44.16 / **40.44**

We further investigate the impact of our two-stage training strategy, which involves the Chart-COT and Chart-RFT stages. By comparing to a baseline without the Chart-COT phase, we analyze the resulting training process, specifically the reward and response length. We find that the first SFT on

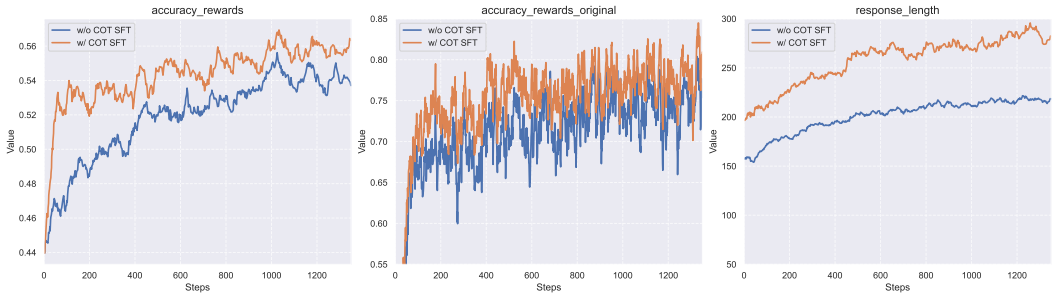


Figure 4: Training curves for the RL stage using the ChartQA and ChartRQA datasets. The orange curve represents our proposed two-stage training strategy, while the blue curve corresponds to a baseline RL-only setting.

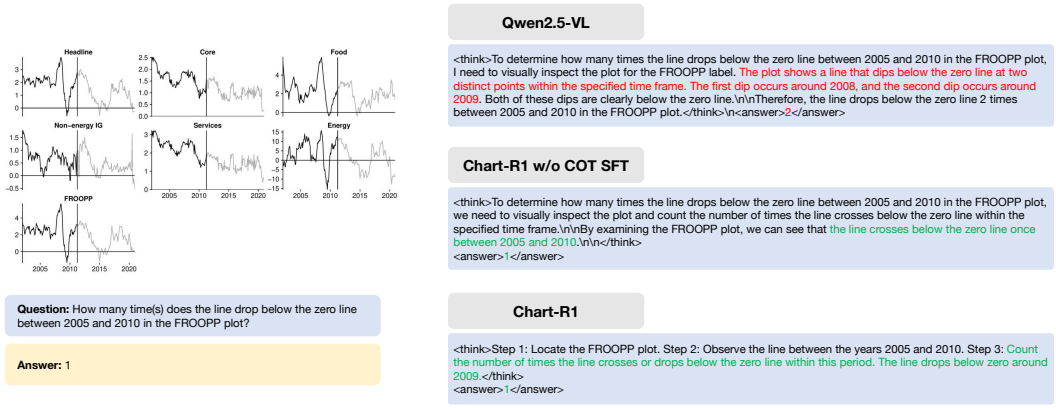


Figure 5: The visualization result of a case study that Chart-R1 (w/ and w/o Chart-COT) answer correctly, but Qwen2.5VL-7B fails.

CoT data has two key benefits. First, it significantly increases the token length generated during the RL phase. Second, it leads to a much effective accuracy reward curve, which rises quickly at the start of training and then converges at a higher final value.

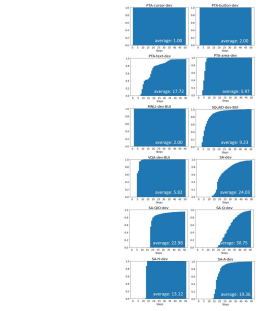
Reward Function For the reward function, we conduct experiments to assess the different settings of the accuracy reward. The results are denoted in Table 5. To better assess the influence caused by different accuracy rewards, we conduct ablation studies that train Qwen2.5VL-7B-Instruct for the RL stage only. The results demonstrate that employing a soft accuracy reward, which combines edit distance for string-based tasks and soft matching for numerical tasks, yields superior performance across the majority of our benchmarks. This finding underscores the importance of adjusting the reward function to the specific type of answers.

SFT Data When training Chart-R1, our SFT dataset consists of 228k samples from our ChartRQA-SFT. We then ablate the SFT data composition by adding two sources, the ChartQA dataset and the 30k ChartRQA-RL that overlaps with the RL data, to assess the impact on performance. We train each setting for 2k steps and 1 epoch for SFT and RL, respectively. The results in Table 6 show that combining ChartQA and ChartRQA-RL, the final performance decreases evidently. Our analysis indicates that using overlapping data for SFT and RL leads to overfitting, where the model memorizes reasoning paths from the SFT stage, resulting in more rigid thinking processes and a significant loss of output diversity. Also, the direct-answer format of the ChartQA data discourages the model from developing the ability to break down problems into a step-by-step thinking process.

Visualization We conclude with qualitative case studies where our Chart-R1 model successfully generates detailed reasoning and correct answers for complex questions in Figures 5 and 6. In these same instances, the baseline Qwen2.5VL-7B model fails, directly demonstrating the superior performance and more advanced reasoning capabilities of our approach. Without training on the Chart-COT stage, Chart-R1 w/o Chart-COT also fails to answer problems in Figure 6, which shows the importance of our proposed two-stage training.

Table 6: Ablation study about the combination of SFT data.

SFT Training Set	ChartQA	CharXiv-RQ	ChartRQA (single / multi)
RQA-SFT	**89.88**	**44.5**	**48.02 / 48.11**
RQA-SFT + QA-Train+ RQA-RL	88.40	41.2	45.98 / 44.60



Question: Which plot has the third highest average value?

Answer: SA-QID-dev

Qwen2.5-VL

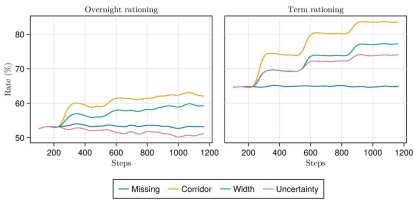
<answer>SA-Q-dev</answer>

Chart-R1 w/o COT SFT

<think>To determine which plot has the third highest average value, we need to compare the averages provided for each plot. The averages are as follows:\n\n- PTA-cursor-dev: 1.00\n- PTA-button-dev: 2.00\n- PTA-text-dev: 17.72\n- PTA-area-dev: 5.97\n- MNLI-dev-BUI: 2.00\n- SQuAD-dev-BUI: 9.23\n- VQA-dev-BUI: 5.82\n- SA-dev: 24.03\n- SA-QID-dev: 22.98\n- SA-Q-dev: 30.75\n- SA-H-dev: 13.12\n- SA-A-dev: 19.36\n\nThe highest averages are:\n1. SA-Q-dev: 30.75\n2. SA-QID-dev: 22.98\n3. SA-dev: 24.03\n\nThe plot with the third highest average value is SA-dev.</think>
<answer>SA-dev</answer>

Chart-R1

<think>Step 1: List all average values from the chart: PTA-cursor-dev (1.00), PTA-button-dev (2.00), PTA-text-dev (17.72), PTA-area-dev (5.97), MNLI-dev-BUI (2.00), SQuAD-dev-BUI (9.23), VQA-dev-BUI (5.82), SA-dev (24.03), SA-QID-dev (22.98), SA-Q-dev (30.75), SA-H-dev (13.12), SA-A-dev (19.36). Step 2: Sort the average values in descending order: 30.75 (SA-Q-dev), 24.03 (SA-dev), 22.98 (SA-QID-dev), 19.36 (SA-A-dev), 17.72 (PTA-text-dev), 13.12 (SA-H-dev), 9.23 (SQuAD-dev-BUI), 5.97 (PTA-area-dev), 5.82 (VQA-dev-BUI), 2.00 (PTA-cursor-dev), 2.00 (PTA-button-dev), 2.00 (MNLI-dev-BUI). Step 3: Identify the third highest average value, which is 22.98. Step 4: Identify the subplot corresponding to this value, which is SA-QID-dev.</think>
<answer>SA-QID-dev</answer>



Question: By the end of the chart Overnight rationing, how many variables remain below 60%?

Answer: 3

Qwen2.5-VL

<think>By the end of the chart for \"Overnight rationing,\" the variables \"Missing\" and \"Uncertainty\" remain below 60%. The \"Corridor\" and \"Width\" variables both reach above 60% by the end of the chart.\n\nSo, the number of variables remaining below 60% is 2.</think>
<answer>2</answer>

Chart-R1 w/o COT SFT

<think>To determine how many variables remain below 60% by the end of the \"Overnight rationing\" chart, we need to examine the lines for each variable and see if they are below the 60% mark at the far right of the chart.\n\n1. **Missing**: The line is around 54%, which is below 60%.\n2. **Corridor**: The line is around 62%, which is above 60%.\n3. **Width**: The line is around 58%, which is below 60%.\n4. **Uncertainty**: The line is around 51%, which is below 60%.\n\nBy the end of the chart, the variables \"Missing\" and \"Width\" are below 60%, while \"Corridor\" and \"Uncertainty\" are above 60%.\n\nTherefore, the number of variables remaining below 60% is 2.</think>
<answer>2</answer>

Chart-R1

<think>Step 1: Examine the 'Overnight rationing' chart. Step 2: Identify the 60% line on the y-axis. Step 3: Observe the data points for each variable at the end of the chart (around 1150 steps). Step 4: 'Missing' is approximately 53.5%, which is below 60%. Step 5: 'Corridor' is approximately 62%, which is not below 60%. Step 6: 'Width' is approximately 59.5%, which is below 60%. Step 7: 'Uncertainty' is approximately 51%, which is below 60%. Step 8: Count the variables that are below 60%: 'Missing', 'Width', and 'Uncertainty'. There are 3 such variables.</think>
<answer>3</answer>

Figure 6: The visualization result of a case study that Chart-R1 (w/ Chart-COT) answers correctly, but Qwen2.5VL-7B and Chart-R1 (w/o Chart-COT) fail.

5 Conclusion

In this paper, we propose Chart-R1, a chart-domain VLM for complex chart reasoning. To improve the reasoning capacity of Chart-R1, we introduce a programmatic data generation method alongside a novel two-stage training strategy to optimize the data construction and training methodology. Also, we propose ChartRQA, which contains 258k training samples which each constructed in verifiable formats and a benchmark for evaluating complex chart reasoning. The result shows that combining our proposed training strategy, Chart-R1 achieves superior performance compared with other VLMs.

A More Training Details

In this section, we provide more details about Chart-R1's training process, including Chart-COT and Chart-RFT.

Chart-COT We use Qwen2.5VL-7B-Instruct as the initial model and perform supervised fine-tuning using LLaMA-Factory Zheng et al. [2024]. We train the model on the 228k ChartRQA-SFT dataset for one epoch. During training, we freeze the vision tower and multi-modal projector parameters and tune the LLM. The learning rate is set to 1e-5, with a warm-up ratio of 0.1 and batch size of 48. The training process costs 3 hours on 24 H800 GPUs.

Chart-RFT For the RFT stage, we use the fine-tuned model from the Chart-COT stage. We adopt the MM-EUREKA Meng et al. [2025a] framework based on OpenRLHF for training. The model is trained for 3 episodes using 30k ChartQA and 30k ChartRQA-RL. We set the rollout batch size and the training batch size to 128, with each sample generating 8 rollouts. The temperature for model generation is set to 1, and we exclude KL divergence in the loss calculation. The learning rate is set to 1e-6, with a warm-up ratio of 0.03, while freezing the vision tower during training. Following the default setting for instruction models, the format reward coefficient is set to 0.5. We employ the online filtering strategy with lower and upper bounds of 0.1 and 0.9, respectively. The training process costs 30 hours on 24 H800 GPUs.

B Benchmark Details

ChartQA Masry et al. [2022] focuses on chart question answering with complex reasoning questions that involve logical and arithmetic operations. Following the settings in the original paper, we evaluate models on the test set reporting overall accuracy scores across both human-written (ChartQA-H) and machine-generated (ChartQA-M) question subsets.

CharXiv Wang et al. [2024] presents a comprehensive evaluation suite with natural, challenging, and diverse charts from arXiv papers to provide a more realistic assessment of chart understanding capabilities. We evaluate models on the Reasoning Questions (CharXiv-RQ) subset, which requires synthesizing information across complex visual elements in charts. Following the original paper, we use GPT-assisted evaluation to assess model responses.

ChartQAPro Masry et al. [2025] introduces a diverse benchmark with various chart types, including infographics and dashboards, and question formats that better reflect real-world challenges. We evaluate models using Chain-of-Thought (CoT) prompting in the original paper and report overall accuracy across five question types.

C ChartRQA Analysis

We count the quantity and distribution of different chart types across the training and test sets of ChartRQA, as detailed in Table A. The distribution among the various types to be well-balanced. Furthermore, Figures A and B provide visualization examples of 24 chart types from the ChartRQA dataset, showcasing both single-chart and multi-chart formats, respectively.

Table A: The detailed chart types and corresponding quantities in our proposed ChartRQA train and test set. ChartRQA contains 24 chart types, each of which contains approximate samples.

Split	Bar	Line	ErrorBar	Heatmap	Box	Scatter	Histogram	Radar	3D
Train	11,850	10,752	11,838	8,993	12,112	10,299	15,856	9,483	9,746
Test	100	88	83	60	103	76	116	46	65
Split	Pie	ErrorPoint	Violin	Area	Bubble	Multi-axes	Ring	Rose	Treemap
Train	17,812	10,814	12,571	9,175	8,996	10,776	12,726	10,533	9,850
Test	103	68	116	75	51	61	54	61	64
Split	Bar_num	Contour	Density	Graph	Quiver	Funnel	Total		
Train	12,150	10,291	12,860	8,764	9,955	227	258,429		
Test	64	67	77	47	52	5	1,702		

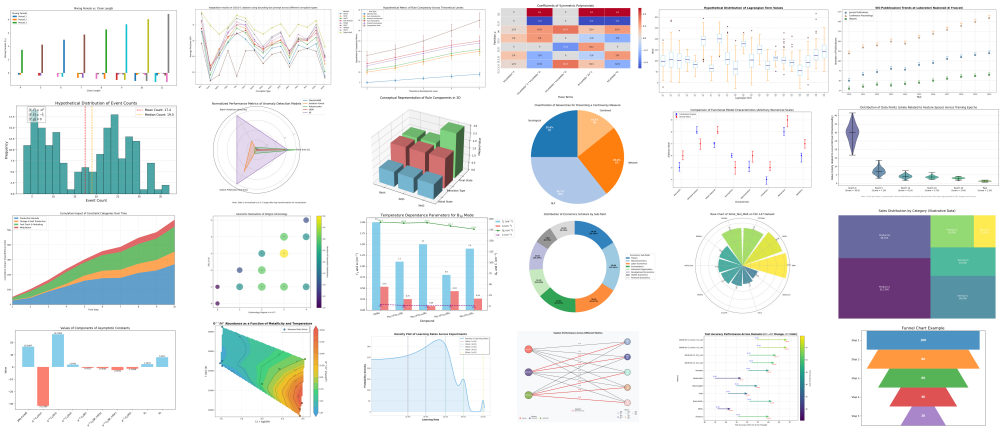


Figure A: Single-chart samples of 24 chart types from ChartRQA.

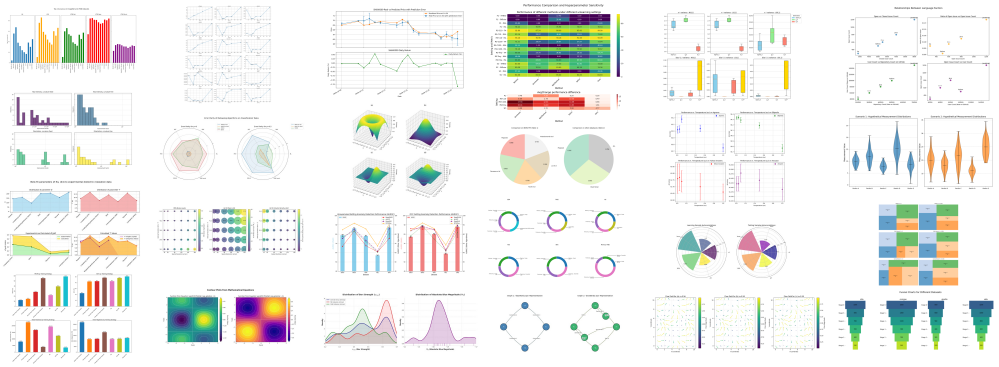


Figure B: Multi-chart samples of 24 chart types from ChartRQA.

D Prompts

To enhance transparency and reproducibility, we provide the exact prompts used for dataset generation and evaluation.

Figure C illustrates the prompt used for code generation. We utilize real table data as input, select one chart type from the 24 predefined chart types, and sample a code example corresponding to that chart type to generate the plotting code for the given table data.

Figures D and E display the prompts used to generate reasoning QA pairs for single-chart and multi-chart formats, respectively. We craft an example for each format to aid LLMs in understanding complex chart reasoning tasks and to generate step-by-step reasoning processes and precise answers that conform to the format. The executable plotting code is provided as auxiliary information to LLMs, making the generated QA pairs more reliable.

Figure F shows the prompt used for model evaluation. We employ GPT-4o to assess the match between the ground truth and the model's predictions, where GPT-4o returns a score of 0 or 1 to indicate the correctness of the model's prediction. Our evaluation focuses solely on the correctness of the final answer, disregarding the reasoning process.

Prompt for Code Generation

Generate high quality python code for plotting {chart_type} chart from the following table data:
{table_data}

Requirements:
The code must present table data in a reasonable way.
The code example of {chart_type} chart (given in JSON format) is:
{code_example}

You must not be limited by the code sample and draw different styles of dials.
The generated code should not be too complicated and all text elements (labels, titles, legends) must be fully visible without overlap or truncation.
Pie/Ring/Treemap chart visualization: always display the actual numerical values on each segment. Percentages are optional, but values must be clearly visible.
IMPORTANT: Generate only ONE figure with all necessary information. If multiple plots are needed, use subplots (plt.subplots) to arrange them in a single figure.
Output format: ```python ... ```

Figure C: Prompt for code generation.

Prompt for Reasoning QA Pairs Generation (Single-chart)

Please propose three questions regarding the input chart image that require strong visual and numerical reasoning skills to answer. These questions should involve multi-step reasoning processes that challenge analytical abilities. Provide detailed answers with step-by-step reasoning. The reasoning process and final answer should be enclosed within <think> and <answer> tags, respectively.

Below is the Python code used to generate this chart. You can use this as reference, but your questions and answers should be based on the visual elements and data actually displayed in the chart image:
```python
{python_code}
```

Guidelines for Effective Reasoning Questions
1. Questions should require 2-5 reasoning steps to solve
2. Include questions about relationships between different data points or series
3. Ask about mathematical operations (differences, percentages, ratios) between data elements
4. Focus on identifying patterns, extremes, or anomalies in the data visualization

Example of a Strong Reasoning Question
Question: What is the sum of the max value of Series A and the min value of Series B?
Answer:
<think>
Step 1: First, identify all values of Series A in the chart. The values are [23, 45, 32, 18, 50].
Step 2: The maximum value of Series A is 50.
Step 3: Next, identify all values of Series B in the chart. The values are [42, 38, 45, 40, 41].
Step 4: The minimum value of Series B is 38.
Step 5: Finally, calculate the sum: 50 + 38 = 88.
</think>
<answer>
88
</answer>

Please strictly adhere to the information displayed in the image when posing questions and providing answers. The answers should be obtainable solely through observation of the image. Avoid posing open-ended questions, and ensure a definite answer using a single word or phrase for each question. Do not fabricate questions or propose questions requiring external knowledge to solve.

Your response should strictly follow the format below and be returned in JSON format:
[{{"Question": "Your first question here...", "Answer": "<think>Your first thinking process here...</think><answer>Your first answer here...</answer>"}}, {{"Question": "Your second question here...", "Answer": "<think>Your second thinking process here...</think><answer>Your second answer here...</answer>"}}, {{"Question": "Your third question here...", "Answer": "<think>Your third thinking process here...</think><answer>Your third answer here...</answer>"}}]

Figure D: Prompt for reasoning QA pairs generation for single-chart formats.

Prompt for Reasoning QA Pairs Generation (Multi-chart)

Please propose three questions regarding the input multi-subplot chart image that require strong cross-subplot visual and numerical reasoning skills to answer. These questions must necessitate analyzing and integrating information from multiple subplots to arrive at the correct answer. Provide detailed answers with step-by-step reasoning processes. The reasoning process and final answer should be enclosed within <think> and <answer> tags, respectively.

Below is the Python code used to generate this multi-subplot chart. You can use this as reference, but your questions and answers should be based on the visual elements and data actually displayed across all subplots in the chart image:
```python
{python_code}
```

Guidelines for Cross-Subplot Questions
1. Each question MUST require information from at least two different subplots to answer correctly
2. Questions should involve comparisons, relationships, or integrations across different subplots
3. Include questions that require mathematical operations (e.g., differences, ratios, correlations) between data from multiple subplots
4. Focus on identifying patterns, trends, or anomalies that are only visible when considering multiple subplots together

Example of Cross-Subplot Question
Question: If we compare the maximum value in subplot A with the average value in subplot B, what is their percentage difference?
Answer:
<think>
Step 1: Identify the maximum value in subplot A. Looking at the first subplot, I can see that the maximum value is 85.
Step 2: Calculate the average value in subplot B. In the second subplot, the values are [42, 38, 45, 40, 41], so the average is (42+38+45+40+41)/5 = 206/5 = 41.2.
Step 3: Calculate the percentage difference: ((85-41.2)/41.2)*100 = (43.8/41.2)*100 = 106.31%
</think>
<answer>
106.31%
</answer>

Please strictly adhere to the information displayed across all subplots when posing questions and providing answers. The answers should be obtainable solely through observation of the image. Avoid posing open-ended questions, and ensure a definite answer using a single word or phrase for each question. Do not fabricate questions or propose questions requiring external knowledge to solve.

Your response should strictly follow the format below and be returned in JSON format:
[{{"Question": "Your first question here...", "Answer": "<think>Your first thinking process here...</think><answer>Your first answer here...</answer>"}}, {{"Question": "Your second question here...", "Answer": "<think>Your second thinking process here...</think><answer>Your second answer here...</answer>"}}, {{"Question": "Your third question here...", "Answer": "<think>Your third thinking process here...</think><answer>Your third answer here...</answer>"}}]

Figure E: Prompt for reasoning QA pairs generation for multi-chart formats.

Prompt for ChartRQA Model Evaluation

You will be given a question, a ground truth answer, and a model response. Your task is to compare the model response with the ground truth answer and assign a binary score (0 or 1). Please provide only the score without any explanations or additional text. If there is no model response provided, assign a score of 0.

Please follow these scoring rules:

Scoring Rules

1. **For Terminology and Concepts:**
* Score 1: The model response and ground truth refer to the same concept or term, even if expressed differently (e.g., a and alpha; $R^2_{t,h,v,m}$ and R^2_t,h,v,m). Different ordering of terms is acceptable when multiple terms are requested.
* Score 0: Any term in the response differs meaningfully from the ground truth (e.g., ACC+ vs ACC; P-101 vs P=101).

Example 1.1:
* Question: What is the name of the curve that intersects y=\lambda exactly three times?
* Ground Truth: P56962
* Response: There is only one curve that intersects y=\lambda exactly three times. The name of the curve is P55762.
Score: 0

Example 1.2:
* Question: What is the letter of the subplot where all bars are above 35?
* Ground Truth: (b)
* Response: The letter of the subplot where all bars are above 35 is b.
Score: 1

2. **For Numerical Values:**
* Score 1: The numerical values in the response and ground truth are mathematically equivalent, even if expressed in different notations (e.g., 0.01 and 10^-2; 1500 and 1.5e3).
* Score 0: The numerical values differ in their actual value, regardless of notation.

Example 2.1:
* Question: What is the value of the red curve at t=10?
* Ground Truth: 0.01
* Response: The value of the red curve at t=10 is 0.012.
Score: 0

Example 2.2:
* Question: What is the value of the blue curve at t=50?
* Ground Truth: 1500
* Response: The value of the blue curve at t=50 is 1.5e3.
Score: 1

3. **For Descriptive Trends and Patterns:**
* Score 1: The response conveys the same semantic meaning as the ground truth (e.g., "increasing then decreasing" and "moving up then down"; "converge" and "move closer together").
* Score 0: The response conveys a different semantic meaning from the ground truth (e.g., "increasing then decreasing" vs "remain constant"; "converge" vs "diverge").

Example 3.1:
* Question: What is the trend of the red curve between t=10 and t=25?
* Ground Truth: increasing then decreasing
* Response: The red curve is increasing between t=10 and t=25.
Score: 0

4. **For Multiple-Choice or Predefined Options:**
* Score 1: The selected option in the response matches the ground truth exactly.
* Score 0: The selected option differs from the ground truth.

Example 4.1:
* Question: What interval among [0, 50], [50, 100], [100, 150], and [150, 200] contains the maximum value of the blue curve?
* Ground Truth: [50, 100]
* Response: The interval where the blue curve achieves the maximum value is [50, 100].
Score: 1

Your Task
* Question: <|question|>
* Ground Truth: <|ground_truth|>
* Response: <|response|>

Score:

Figure F: Prompt for ChartRQA evaluation using GPT-4o.

References

M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl, A. Benhaim, M. Bilenko, J. Bjorck, S. Bubeck, M. Cai, Q. Cai, V. Chaudhary, D. Chen, D. Chen, W. Chen, Y.-C. Chen, Y.-L. Chen, H. Cheng, P. Chopra, X. Dai, M. Dixon, R. Eldan, V. Fragoso, J. Gao, M. Gao, M. Gao, A. Garg, A. D. Giorno, A. Goswami, S. Gunasekar, E. Haider, J. Hao, R. J. Hewett, W. Hu, J. Huynh, D. Iter, S. A. Jacobs, M. Javaheripi, X. Jin, N. Karampatziakis, P. Kauffmann, M. Khademi, D. Kim, Y. J. Kim, L. Kurilenko, J. R. Lee, Y. T. Lee, Y. Li, Y. Li, C. Liang, L. Liden, X. Lin, Z. Lin, C. Liu, L. Liu, M. Liu, W. Liu, X. Liu, C. Luo, P. Madan, A. Mahmoudzadeh, D. Majercak, M. Mazzola, C. C. T. Mendes, A. Mitra, H. Modi, A. Nguyen, B. Norick, B. Patra, D. Perez-Becker, T. Portet, R. Pryzant, H. Qin, M. Radmilac, L. Ren, G. de Rosa, C. Rosset, S. Roy, O. Ruwase, O. Saarikivi, A. Saied, A. Salim, M. Santacroce, S. Shah, N. Shang, H. Sharma, Y. Shen, S. Shukla, X. Song, M. Tanaka, A. Tupini, P. Vaddamanu, C. Wang, G. Wang, L. Wang, S. Wang, X. Wang, Y. Wang, R. Ward, W. Wen, P. Witte, H. Wu, X. Wu, M. Wyatt, B. Xiao, C. Xu, J. Xu, W. Xu, J. Xue, S. Yadav, F. Yang, J. Yang, Y. Yang, Z. Yang, D. Yu, L. Yuan, C. Zhang, C. Zhang, J. Zhang, L. L. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, and X. Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL `https://arxiv.org/abs/2404.14219`.

Anthropic. Introducing the next generation of claude, 2024. URL `https://www.anthropic.com/news/claude-3-family`.

S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

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

D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Y. Han, C. Zhang, X. Chen, X. Yang, Z. Wang, G. Yu, B. Fu, and H. Zhang. Chartllama: A multimodal llm for chart understanding and generation. *arXiv preprint arXiv:2311.16483*, 2023.

W. He, Z. Xi, W. Zhao, X. Fan, Y. Ding, Z. Shan, T. Gui, Q. Zhang, and X. Huang. Distill visual chart reasoning ability from llms to mllms. *arXiv preprint arXiv:2410.18798*, 2024.

L. Hu, D. Wang, Y. Pan, J. Yu, Y. Shao, C. Feng, and L. Nie. 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*, pages 3917–3925, 2024.

W. Huang, B. Jia, Z. Zhai, S. Cao, Z. Ye, F. Zhao, Z. Xu, Y. Hu, and S. Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025.

C. Jia, N. Xu, J. Wei, Q. Wang, L. Wang, B. Yu, and J. Zhu. Chartreasoner: Code-driven modality bridging for long-chain reasoning in chart question answering. *arXiv preprint arXiv:2506.10116*, 2025.

S. Kantharaj, R. T. K. Leong, X. Lin, A. Masry, M. Thakkar, E. Hoque, and S. Joty. Chart-to-text: A large-scale benchmark for chart summarization. *arXiv preprint arXiv:2203.06486*, 2022.

F. Liu, X. Wang, W. Yao, J. Chen, K. Song, S. Cho, Y. Yacoob, and D. Yu. Mmc: Advancing multimodal chart understanding with large-scale instruction tuning. *arXiv preprint arXiv:2311.10774*, 2023.

Z. Liu, C. Chen, W. Li, P. Qi, T. Pang, C. Du, W. S. Lee, and M. Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

A. Masry, D. X. Long, J. Q. Tan, S. Joty, and E. Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

A. Masry, M. Shahmohammadi, M. R. Parvez, E. Hoque, and S. Joty. Chartinstruct: Instruction tuning for chart comprehension and reasoning. *arXiv preprint arXiv:2403.09028*, 2024a.

A. Masry, M. Thakkar, A. Bajaj, A. Kartha, E. Hoque, and S. Joty. Chartgemma: Visual instruction-tuning for chart reasoning in the wild. *arXiv preprint arXiv:2407.04172*, 2024b.

A. Masry, M. S. Islam, M. Ahmed, A. Bajaj, F. Kabir, A. Kartha, M. T. R. Laskar, M. Rahman, S. Rahman, M. Shahmohammadi, et al. Chartqapro: A more diverse and challenging benchmark for chart question answering. *arXiv preprint arXiv:2504.05506*, 2025.

F. Meng, L. Du, Z. Liu, Z. Zhou, Q. Lu, D. Fu, T. Han, B. Shi, W. Wang, J. He, K. Zhang, P. Luo, Y. Qiao, Q. Zhang, and W. Shao. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025a.

F. Meng, L. Du, Z. Liu, Z. Zhou, Q. Lu, D. Fu, T. Han, B. Shi, W. Wang, J. He, et al. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025b.

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

M. Ni, Z. Yang, L. Li, C.-C. Lin, K. Lin, W. Zuo, and L. Wang. Point-rft: Improving multimodal reasoning with visually grounded reinforcement finetuning. *arXiv preprint arXiv:2505.19702*, 2025.

OpenAI. Gpt-4o, 2024. URL `https://openai.com/index/hello-gpt-4o`. Accessed: 2024-05-13.

OpenAI. Introducing openai o3 and o4-mini. `https://openai.com/index/introducing-o3-and-o4-mini/`, April 2025. Accessed: 2025-07-14.

H. Qiu, X. Lan, F. Liu, X. Sun, D. Ruan, P. Shi, and L. Ma. Metis-rise: Rl incentivizes and sft enhances multimodal reasoning model learning. *arXiv preprint arXiv:2506.13056*, 2025.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

H. Shen, P. Liu, J. Li, C. Fang, Y. Ma, J. Liao, Q. Shen, Z. Zhang, K. Zhao, Q. Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

C. Shi, C. Yang, Y. Liu, B. Shui, J. Wang, M. Jing, L. Xu, X. Zhu, S. Li, Y. Zhang, et al. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation. *arXiv preprint arXiv:2406.09961*, 2024.

A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.

L. Tang, G. Kim, X. Zhao, T. Lake, W. Ding, F. Yin, P. Singhal, M. Wadhwa, Z. L. Liu, Z. Sprague, et al. Chartmuseum: Testing visual reasoning capabilities of large vision-language models. *arXiv preprint arXiv:2505.13444*, 2025.

G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Y. Wang, S. Wu, Y. Zhang, S. Yan, Z. Liu, J. Luo, and H. Fei. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*, 2025.

Z. Wang, M. Xia, L. He, H. Chen, Y. Liu, R. Zhu, K. Liang, X. Wu, H. Liu, S. Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *arXiv preprint arXiv:2406.18521*, 2024.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Y. Wu, L. Yan, L. Shen, Y. Mei, J. Wang, and Y. Luo. Chartcards: A chart-metadata generation framework for multi-task chart understanding. *arXiv preprint arXiv:2505.15046*, 2025.

Z. Wu, X. Chen, Z. Pan, X. Liu, W. Liu, D. Dai, H. Gao, Y. Ma, C. Wu, B. Wang, Z. Xie, Y. Wu, K. Hu, J. Wang, Y. Sun, Y. Li, Y. Piao, K. Guan, A. Liu, X. Xie, Y. You, K. Dong, X. Yu, H. Zhang, L. Zhao, Y. Wang, and C. Ruan. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding, 2024. URL `https://arxiv.org/abs/2412.10302`.

R. Xia, B. Zhang, H. Ye, X. Yan, Q. Liu, H. Zhou, Z. Chen, M. Dou, B. Shi, J. Yan, et al. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *arXiv preprint arXiv:2402.12185*, 2024.

Z. Xu, B. Qu, Y. Qi, S. Du, C. Xu, C. Yuan, and J. Guo. Chartmoe: Mixture of diversely aligned expert connector for chart understanding. *arXiv preprint arXiv:2409.03277*, 2024.

Y. Yang, X. He, H. Pan, X. Jiang, Y. Deng, X. Yang, H. Lu, D. Yin, F. Rao, M. Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*, 2025.

L. Zhang, A. Hu, H. Xu, M. Yan, Y. Xu, Q. Jin, J. Zhang, and F. Huang. Tinychart: Efficient chart understanding with visual token merging and program-of-thoughts learning. *arXiv preprint arXiv:2404.16635*, 2024.

X. Zhao, X. Liu, H. Yang, X. Luo, F. Zeng, J. Li, Q. Shi, and C. Chen. Chartedit: How far are mllms from automating chart analysis? evaluating mllms' capability via chart editing. *arXiv preprint arXiv:2505.11935*, 2025a.

X. Zhao, X. Luo, Q. Shi, C. Chen, S. Wang, Z. Liu, and M. Sun. Chartcoder: Advancing multimodal large language model for chart-to-code generation. *arXiv preprint arXiv:2501.06598*, 2025b.

Y. Zheng, R. Zhang, J. Zhang, Y. Ye, Z. Luo, Z. Feng, and Y. Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL `http://arxiv.org/abs/2403.13372`.

J. Zhu, W. Wang, Z. Chen, Z. Liu, S. Ye, L. Gu, H. Tian, Y. Duan, W. Su, J. Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.