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

Anonymous ACL submission

Abstract

Chart reasoning presents unique challenges due to its inherent complexity—requiring precise numerical comprehension, multi-level visual understanding, and logical inference across interconnected data elements. Existing vision-language models often struggle with such reasoning tasks, particularly when handling multi-subchart scenarios and numerical sensitivity. To address these challenges, we introduce **Chart-R1**, a chart-domain vision-language model that leverages reinforcement fine-tuning for advanced chart reasoning. We first propose a programmatic data synthesis approach to generate high-quality step-by-step reasoning data with verifiable answer formats, covering diverse chart types and complexity levels. Our two-stage training strategy includes: (1) **Chart-COT**, which decomposes complex reasoning into interpretable subtasks through chain-of-thought supervision, and (2) **Chart-RFT**, which employs group relative policy optimization with numerically sensitive rewards tailored for chart-specific reasoning. Experiments on open-source benchmarks and our proposed **ChartRQA** dataset demonstrate that Chart-R1 significantly outperforms existing chart-domain methods and rivals large-scale open/closed-source models.

1 Introduction

Recently, inspired by the success of models such as OpenAI's o1/o3 (OpenAI, 2025b) 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). Chart reasoning requires decomposing complex questions into multiple interpretable steps involving precise numerical comprehension, visual parsing, and logical inference. As illustrated in Figure 5, questions with fewer reasoning steps are relatively straightforward, but the difficulty increases substantially when problems require multi-step decomposition. Prior works attempt to improve chart reasoning through 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). However, SFT causes models to overfit specific reasoning patterns, hindering their generalization abilities. Recent methods (Jia et al., 2025; Ni et al., 2025) leverage RL to enhance VLM reasoning capabilities, yet their focus remains primarily on perception and understanding rather than the multi-step reasoning required for complex chart analysis.

In this work, we propose Chart-R1, a chart-domain VLM that leverages RL to enhance advanced multi-step reasoning capability, which achieves superior performance as shown in Figure 1. To support Chart-R1, we introduce two key contributions. First, we propose a programmatic synthesis strategy to generate high-quality step-by-step reasoning data with verifiable answer formats, covering diverse chart types and complexity levels. Specifically, we utilize LLMs to generate the chart plotting code from real-world tables curated from arXiv papers, ensuring data fidelity in the resulting charts. The generated code then serves as a foundation for formulating complex questions, multi-step CoT reasoning processes, and verifiable answers,

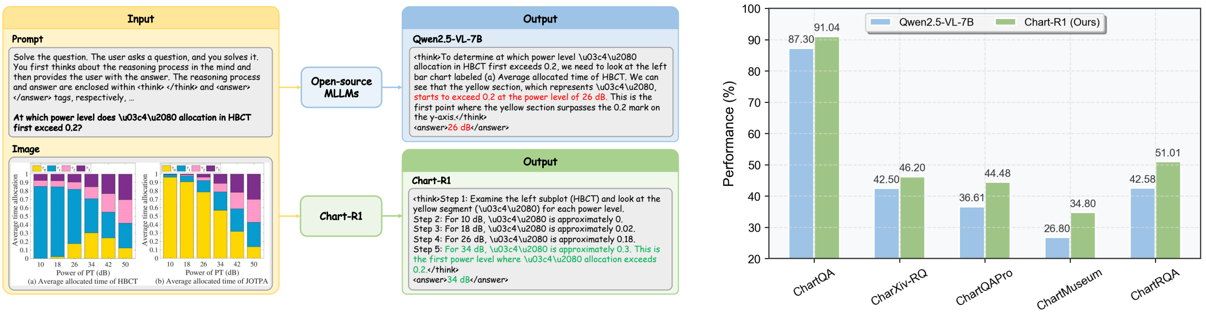

Figure 1: Comparison of Qwen2.5-VL-7B and Chart-R1 on chart understanding and reasoning benchmarks. In the complex chart reasoning task, Qwen2.5-VL-7B generates a wrong thinking process, whereas Chart-R1 thinks and answers correctly.

resulting in ChartRQA—a comprehensive dataset featuring 258k multi-step reasoning samples covering both single- and multi-chart scenarios. Second, we develop an effective two-stage training strategy: (1) Chart-COT with step-by-step chain-of-thought supervision, and (2) Chart-RFT with numerically sensitive reinforcement fine-tuning. During the Chart-COT stage, the model learns to decompose complex tasks into fine-grained, interpretable subtasks through supervised learning on step-by-step reasoning data. As shown in Figure 1, this explicit decomposition capability is crucial for handling complex scenarios that require multi-step reasoning. In the Chart-RFT stage, we employ group relative policy optimization (GRPO) with a composite reward signal of soft matching and edit distance to enhance accuracy for both numerical and string-based answers. We utilize distinct datasets for these two stages, as our findings show that training on the same data impairs the model's exploration ability during the RL process. Furthermore, we introduce ChartRQA, a human-verified benchmark designed to probe the limits of complex chart reasoning. In contrast to prior works (Wang et al., 2024), its questions demand higher complexity and multi-step thought processes. The substantial performance drop of existing VLMs on ChartRQA reveals a critical gap in their reasoning capabilities. In summary, our contributions are as follows:

- To enhance chart reasoning in VLMs, we propose a novel two-stage training strategy consisting of Chart-COT and Chart-RFT. The resulting model, Chart-R1, sets a new state-of-the-art across various chart understanding and reasoning benchmarks.

- We introduce a programmatic data synthesis strategy that leverages code generation from real-world tables to produce step-by-step reasoning data with verifiable answers, ensuring both data fidelity and reasoning quality.

- We present ChartRQA, a comprehensive dataset with a human-verified benchmark and 258k training samples. Existing VLMs show significant performance gaps on this benchmark, revealing critical reasoning limitations.

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 concentrate on descriptive tasks (Masry et al., 2024a,b), such as extracting explicit content from charts (Masry et al., 2022). More recent works focus on leveraging reasoning capabilities to interpret complex and implicit information. For example, TinyChart (Zhang et al., 2024) utilizes a template-based method to generate Program-of-Thought (PoT) (Chen et al., 2022) reasoning data. ChartCoder (Zhao et al., 2025b) proposes Snippet-of-Thought for chart-to-code generation. ChartReasoner (Jia et al., 2025) utilizes a chart-to-code model to convert images into code and generate reasoning processes. However, the generated reasoning data has limitations due to 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 attempt to enhance 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., 2025b; 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 long reasoning. MMEureka (Meng et al., 2025b) and R1-Zero (Liu et al., 2025) further advance visual long-term reasoning with improved RL training strategies. Point-RFT (Ni et al., 2025) uses grounded CoT for visual understanding, but it only utilize ChartQA for RL which limits the model's 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 models' capacity on extracting information from charts, where the challenge is predominantly driven by chart complexity. Recent works such as Charxiv (Wang et al., 2024) and CharMuseum (Tang et al., 2025a) introduce more challenging reasoning tasks that require models to think before answering. Unlike descriptive tasks, reasoning tasks present a dual challenge from both the perceptual complexity of charts 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 like distilling reasoning from top VLMs, so it naturally inherits their limitations and errors on complex tasks. The reliance on final answer correctness as the only supervisory signal makes generating high-quality CoT reasoning data a significant challenge. This issue is amplified in complex chart reasoning, where the struggles of even top models inherently lead to low-quality, undiverse data. 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 are 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 LLM to generate plotting code. This code then serves as a perfect, high-fidelity foundation from which a VLM subsequently synthesizes question-answer pairs and their complex step-by-step reasoning path. An overview of our data synthesis pipeline is shown in Figure 2.

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 enhance diversity, we categorize the plotting code into single- and multi-chart types and apply distinct generation instructions for each. For multi-chart problems, we instruct the

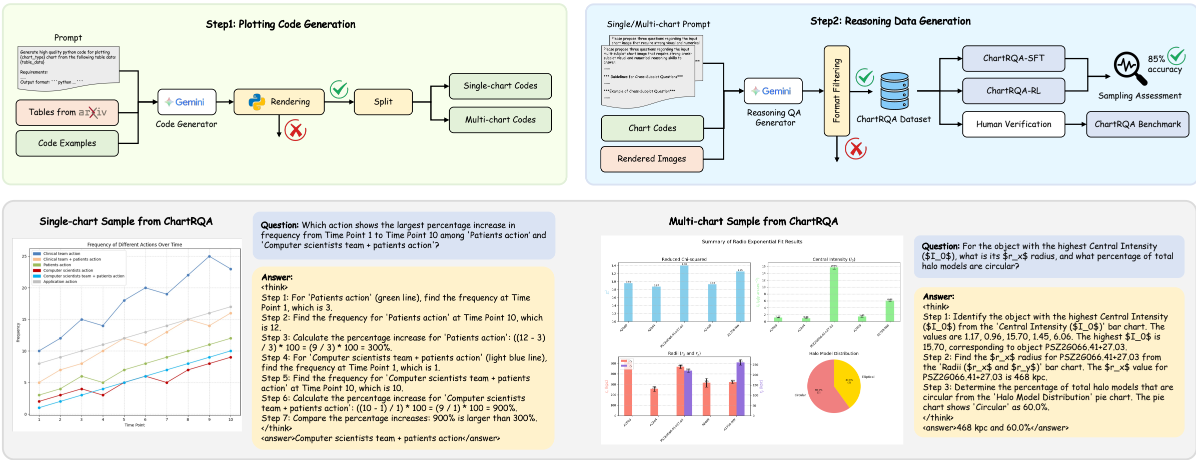

Figure 2: ChartRQA dataset pipeline via programmatic data synthesis. The pipeline consists of plotting code generation and reasoning data generation, producing ChartRQA-SFT, ChartRQA-RL, and the ChartRQA benchmark. Examples show single-chart and multi-chart scenarios requiring multi-step reasoning.

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	✔	✔

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.

Token Avg.	Train		Test	
	Single	Multi	Single	Multi
Question	30.03	39.84	29.83	39.49
Thinking Process	196.50	237.38	196.32	240.94
Answer	5.98	8.87	5.96	8.97

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.

LLM to generate questions that necessitate cross-referencing information between sub-charts. The generated data 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 while enabling the scalable synthesis of data independent of existing corpora.

Dataset Construction Using the aforementioned methods, we construct the ChartRQA dataset, which 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 sets 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.

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. Using Qwen2.5-VL-7B-Instruct as the baseline model, we first fine-tune it via SFT on the step-by-step reasoning data of our proposed ChartRQA-SFT. This initial stage equips the model with the fundamental capability to decompose complex tasks into fine-grained subtasks. Our ablation studies demonstrate that this preliminary SFT stage on CoT data is critical, as it yields significantly better performance than applying RL from scratch. The model is trained using a standard autoregressive language modeling objective with negative log-likelihood loss.

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 this degradation, we subsequently apply reinforcement fine-tuning (RFT) to generalize its reasoning capacity.

Group Relative Policy Optimization We adapt the Group Relative Policy Optimization (GRPO) algorithm (Shao et al., 2024; Guo et al., 2025), which significantly conserves training resources by replacing the critic model with a baseline estimated from group scores. For each input (x, y), the policy π_θ samples a group of G candidate responses $\{o_i\}_{i=1}^{G}$.

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

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

$$A_i = \frac{r_i - \text{mean}\left(\{r_1, r_2, \cdots, r_G\}\right)}{\text{std}\left(\{r_1, r_2, \cdots, r_G\}\right)} \tag{2}$$

Reward Design For effective RFT, we follow the DeepSeek-R1 (Shao et al., 2024) and adopt a rule-based reward function that consists of accuracy and format rewards to assess answer correctness and structural integrity, respectively.

- **Accuracy Reward.** We employ distinct, type-specific reward functions to measure the correctness of model outputs. For numerical answers, we adopt the soft matching technique from Point-RFT (Ni et al., 2025) with 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 by a grammar-level regex parser that validates the structural integrity of outputs. It confirms two conditions: (1) the reasoning process is properly enclosed in <think> tags, and (2) the final answer is extractable from the designated <answer> tags.

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 Experiment Settings

We conduct experiments and ablation studies to evaluate the results obtained from various training settings. See the appendix for training details.

Benchmarks To comprehensively evaluate the understanding and reasoning capacity of Chart-R1, we choose ChartQA (Masry et al., 2022), CharXiv-RQ (Reasoning Questions) (Wang et al., 2024), ChartQAPro (Masry et al., 2025a), ChartMuseum (Tang et al., 2025a) and our proposed ChartRQA (single/multi) as the evaluation benchmarks.

Model Name	ChartQA	CharXiv-RQ	ChartQAPro	ChartMuseum	ChartRQA (single / multi)
			Proprietary		
GPT-4o	85.7	47.1	37.67	42.2	44.37 / 46.55
Gemini-1.5-Flash	79.0	33.9	42.96	31.1	-
Gemini-1.5-Pro	87.2	43.3	-	41.3	-
Gemini-2.5-Flash	-	-	-	-	59.12 / 59.17
Claude-3.5-Sonnet	90.8	60.2	43.58	54.4	52.79 / 56.05
GPT-4.1	86.8	56.7	-	48.4	57.88 / 59.30
Claude-3.7-Sonnet	86.1	64.2	-	60.3	55.04 / 57.87
			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	-	28.2	37.51 / 31.73
InternVL3-38B	89.2	46.4	-	32.1	46.09 / 38.36
Qwen2.5-VL-7B	87.3	42.5	36.61	26.8	44.59 / 40.57
R1-VL	83.9	32.0	35.11	21.0	27.33 / 21.72
VL-Rethinker	83.5	42.8	**48.26**	31.5	42.87 / 42.52
MMR1	83.7	40.5	45.72	30.9	47.27 / 46.55
			Chart-domain		
ChartLlama	69.66	14.2	-	-	-
TinyChart	83.60	8.3	13.25	12.5	6.75 / 6.11
ChartGemma	80.16	12.5	6.84	12.2	7.18 / 9.23
ChartReasoner	86.93	-	39.97	-	-
BigCharts-R1	89.84	41.3	-	-	-
Bespoke-MiniChart	89.50	45.4	45.36	34.0	42.77 / 42.13
Chart-R1 (Ours)	**91.04**	**46.2**	44.48	**34.8**	**52.09 / 49.93**

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

Baselines We compare our proposed Chart-R1 with existing models in three setups: (1) Proprietary models include GPT-4o, GPT-4.1 (OpenAI, 2025a), Gemini-1.5-(Flash, Pro), Gemini-2.5-Flash (Kavukcuoglu, 2025), Claude-3.5-Sonnet and Claude-3.7-Sonnet (Anthropic, 2025). (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); and VLM reasoning models, including R1-VL (Zhang et al., 2025), VL-Rethinker (Wang et al., 2025a) and MMR1 (Leng et al., 2025). (3) Chart-domain VLMs including ChartLlama (Han et al., 2023), TinyChart (Zhang et al., 2024), ChartGemma (Masry et al., 2024b), ChartResoner (Jia et al., 2025), BigCharts-R1 (Masry et al., 2025b) and Bespoke-MiniChart-7B (Tang et al., 2025b).

4.2 Main Results

Table 3 shows the performance of Chart-R1 compared with other baseline models. The results show that Chart-R1 achieves the state-of-the-art performance on small-scale (<20B) VLMs, including general- and chart-domain models across various chart understanding and reasoning 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, ChartMuseum 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 effectiveness of our proposed ChartRQA dataset and CoT-RL training strategy.

4.3 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 yields the most balanced performance. Notably, omitting Chart-COT causes a significant performance drop on the ChartRQA benchmark. We attribute this to 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 decomposi-

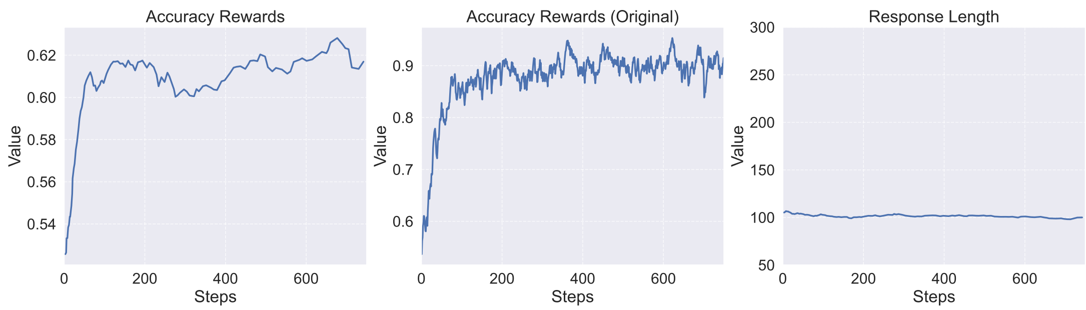

Figure 3: The training curve during the RL stage that utilizes the ChartQA dataset solely.

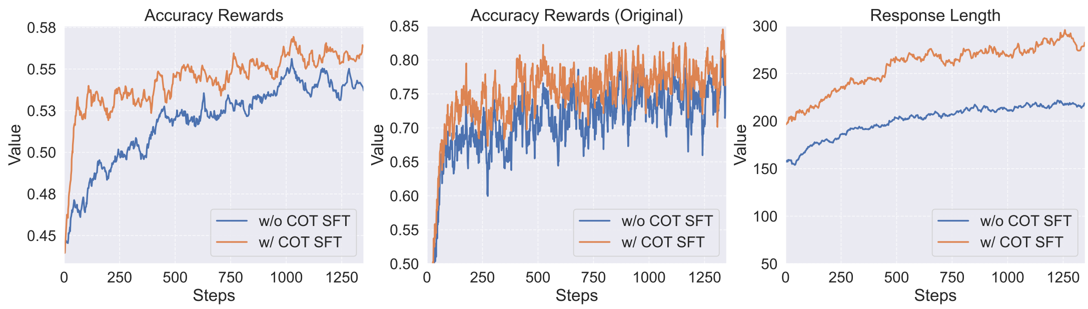

Figure 4: Training curves for the RL stage using the ChartQA and ChartRQA datasets. The orange curve represents our proposed two-stage training strategy, while the blue curve corresponds to a RL-only baseline.

tion. Also, SFT 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.

Prior research underscores the critical role of training data complexity for effective RL (Guo et al., 2025). Our generated ChartRQA training set addresses this by featuring tasks with both single- and multi-chart images, and questions demanding 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 observed in our Chart-RFT stage. Furthermore, RL exclusively on the ChartQA dataset is insufficient for developing a 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.

We further investigate the impact of our two-stage training strategy, comparing it to a baseline without the Chart-COT phase. The comparison of RL processes is shown in Figure 4. We find that the first SFT on 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.

Visualization We present qualitative case studies where our Chart-R1 model successfully generates detailed reasoning and correct answers for complex questions in Figure 5. In these same instances, the baseline Qwen2.5-VL-7B model fails, directly demonstrating the superior performance and more advanced reasoning capabilities of our approach. When Chart-R1 is trained without the Chart-COT stage, it also fails to answer the problems in the right case of Figure 5. Although it can correctly recognize the chart content, it makes errors during the reasoning process, highlighting the importance of our proposed two-stage training.

4.4 Error Analysis

Chart-R1 achieves significant improvements in reasoning ability compared to baseline, but there is still room for further improvement. We randomly sample 50 incorrect responses from Chart-R1 on ChartQAPro and analyze the error types. As shown in Figure 6, Chart-R1 is most prone to errors in vi-

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**

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

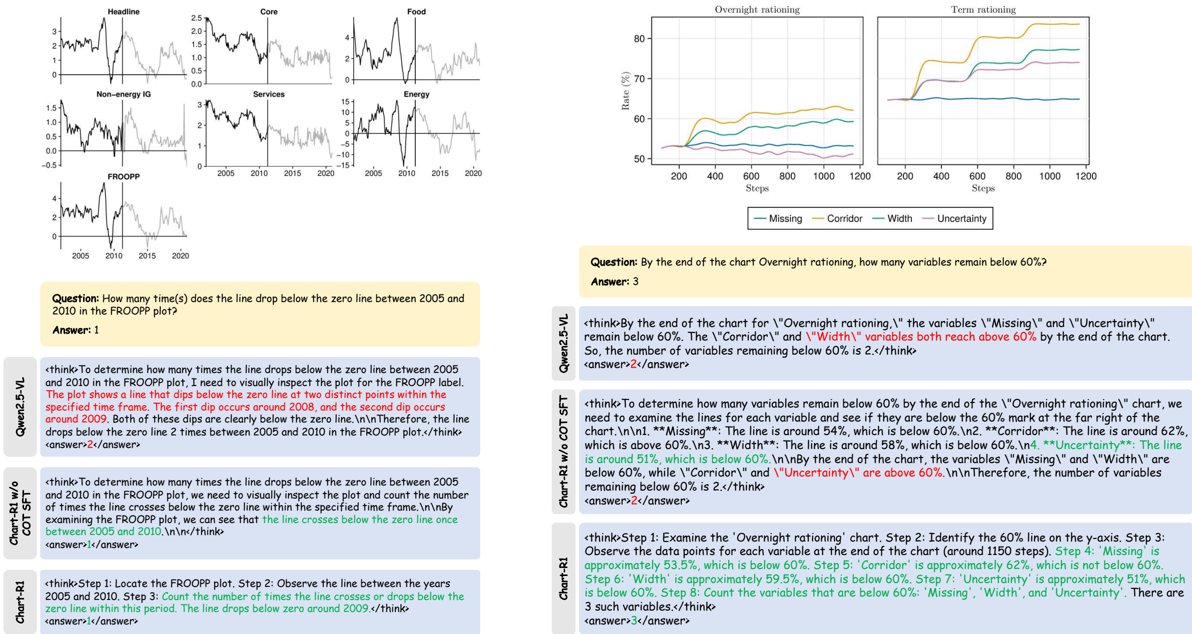

Figure 5: Visualization results of two case studies: (a) Chart-R1 (with and without Chart-COT) both answer correctly while Qwen2.5-VL-7B fails, and (b) only Chart-R1 with Chart-COT answers correctly while both Qwen2.5-VL-7B and Chart-R1 without Chart-COT fail.

sual reasoning, multi-chart QA, and unanswerable types. Visual reasoning is more challenging than mathematical reasoning, as the latter mainly involves numerical recognition and calculation, while the former requires the model to identify and summarize complex chart patterns. Multi-chart QA requires the model to integrate information across multiple charts. While ChartRQA was designed to address multi-chart reasoning, the current model still exhibits deficiencies in this aspect. For unanswerable questions, although ChartRQA did not specifically include such samples, Chart-R1 can reject most unanswerable questions through RL, demonstrating good generalizability.

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, 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.

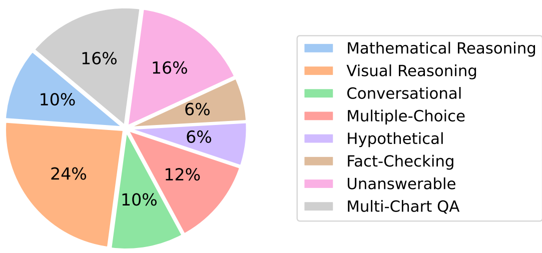

Figure 6: Error type distribution of Chart-R1 on ChartQAPro.

Limitations

Our study focuses primarily on statistical charts from academic papers, overlooking practical visualization types such as dashboards and flowcharts. This leads to a gap compared to closed-source models on comprehensive benchmarks such as ChartMuseum. In future research, we plan to expand our training paradigm to incorporate diverse chart types and complex visual reasoning, developing a more versatile chart understanding model.

Ethical Statement

Our research employs publicly available models and datasets with proper citations. This approach leverages the widespread use and non-toxic nature of our datasets and prompts to minimize the risk of generating toxic content.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Anthropic. 2025. Claude 3.7 sonnet and claude code.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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

Wei He, Zhiheng Xi, Wanxu Zhao, Xiaoran Fan, Yiwen Ding, Zifei Shan, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Distill visual chart reasoning ability from llms to mllms. *arXiv preprint arXiv:2410.18798*.

Linmei Hu, Duokang Wang, Yiming Pan, Jifan Yu, Yingxia Shao, Chong Feng, and Liqiang Nie. 2024. Novachart: A large-scale dataset towards chart understanding and generation of multimodal large language models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3917–3925.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

Caijun Jia, Nan Xu, Jingxuan Wei, Qingli Wang, Lei Wang, Bihui Yu, and Junnan Zhu. 2025. Chartreasoner: Code-driven modality bridging for long-chain reasoning in chart question answering. *arXiv preprint arXiv:2506.10116*.

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

Koray Kavukcuoglu. 2025. Gemini 2.5: Our most intelligent ai model.

Sicong Leng, Jing Wang, Jiaxi Li, Hao Zhang, Zhiqiang Hu, Boqiang Zhang, Yuming Jiang, Hang Zhang, Xin Li, Lidong Bing, and 1 others. 2025. Mmr1: Enhancing multimodal reasoning with variance-aware sampling and open resources. *arXiv preprint arXiv:2509.21268*.

Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. 2023. Mmc: Advancing multimodal chart understanding with large-scale instruction tuning. *arXiv preprint arXiv:2311.10774*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Ahmed Masry, Mohammed Saidul Islam, Mahir Ahmed, Aayush Bajaj, Firoz Kabir, Aaryaman Kartha, Md Tahmid Rahman Laskar, Mizanur Rahman, Shadikur Rahman, Mehrad Shahmohammadi, and 1 others. 2025a. Chartqapro: A more diverse and challenging benchmark for chart question answering. *arXiv preprint arXiv:2504.05506*.

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

Ahmed Masry, Abhay Puri, Masoud Hashemi, Juan A Rodriguez, Megh Thakkar, Khyati Mahajan, Vikas Yadav, Sathwik Tejaswi Madhusudhan, Alexandre Piché, Dzmitry Bahdanau, and 1 others. 2025b. Bigcharts-r1: Enhanced chart reasoning with visual reinforcement finetuning. *arXiv preprint arXiv:2508.09804*.

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

Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. 2024b. Chartgemma: Visual instruction-tuning for chart reasoning in the wild. *arXiv preprint arXiv:2407.04172*.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, Ping Luo, Yu Qiao, Qiaosheng Zhang, and Wenqi Shao. 2025a. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, and 1 others. 2025b. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*.

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

Minheng Ni, Zhengyuan Yang, Linjie Li, Chung-Ching Lin, Kevin Lin, Wangmeng Zuo, and Lijuan Wang. 2025. Point-rft: Improving multimodal reasoning with visually grounded reinforcement finetuning. *arXiv preprint arXiv:2505.19702*.

OpenAI. 2025a. Introducing GPT-4.1 in the API.

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

Haibo Qiu, Xiaohan Lan, Fanfan Liu, Xiaohu Sun, Delian Ruan, Peng Shi, and Lin Ma. 2025. Metis-rise: Rl incentivizes and sft enhances multimodal reasoning model learning. *arXiv preprint arXiv:2506.13056*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, and 1 others. 2025. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*.

Chufan Shi, Cheng Yang, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, and 1 others. 2024. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation. *arXiv preprint arXiv:2406.09961*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Liyan Tang, Grace Kim, Xinyu Zhao, Thom Lake, Wenxuan Ding, Fangcong Yin, Prasann Singhal, Manya Wadhwa, Zeyu Leo Liu, Zayne Sprague, and 1 others. 2025a. Chartmuseum: Testing visual reasoning capabilities of large vision-language models. *arXiv preprint arXiv:2505.13444*.

Liyan Tang, Shreyas Pimpalgaonkar, Kartik Sharma, Alexandros G. Dimakis, Maheswaran Sathiamoorthy, and Greg Durrett. 2025b. Bespoke-minichart-7b: pushing the frontiers of open vlms for chart understanding. https://www.bespokelabs.ai/blog/bespoke-minichart-7b. Accessed: 2025-04-23.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. 2025a. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*.

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. 2025b. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, and 1 others. 2024. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *arXiv preprint arXiv:2406.18521*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yifan Wu, Lutao Yan, Leixian Shen, Yinan Mei, Jiannan Wang, and Yuyu Luo. 2025. Chartcards: A chart-metadata generation framework for multi-task chart understanding. *arXiv preprint arXiv:2505.15046*.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, Zhenda Xie, Yu Wu, Kai Hu, Jiawei Wang, Yaofeng Sun, Yukun Li, Yishi Piao, Kang Guan, Aixin Liu, and 8 others. 2024. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *Preprint*, arXiv:2412.10302.

Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Min Dou, Botian Shi, Junchi Yan, and 1 others. 2024. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *arXiv preprint arXiv:2402.12185*.

Zhengzhuo Xu, Bowen Qu, Yiyan Qi, Sinan Du, Chengjin Xu, Chun Yuan, and Jian Guo. 2024. Chartmoe: Mixture of diversely aligned expert connector for chart understanding. *arXiv preprint arXiv:2409.03277*.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, and 1 others. 2025. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. 2025. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *arXiv preprint arXiv:2503.12937*.

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

Xuanle Zhao, Xuexin Liu, Haoyue Yang, Xianzhen Luo, Fanhu Zeng, Jianling Li, Qi Shi, and Chi Chen. 2025a. Chartedit: How far are mllms from automating chart analysis? evaluating mllms' capability via chart editing. *arXiv preprint arXiv:2505.11935*.

Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2025b. Chartcoder: Advancing multimodal large language model for chart-to-code generation. *arXiv preprint arXiv:2501.06598*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. 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. Association for Computational Linguistics.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Appendix

A.1 Training Details

Chart-COT We use Qwen2.5-VL-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.

A.2 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., 2025a) 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.

ChartMuseum (Tang et al., 2025a) is a chart question-answering benchmark designed to evaluate complex visual reasoning capabilities with expert-annotated questions from diverse real-world charts. Following the original paper, we evaluate models using the provided CoT prompt and LLM-as-a-Judge evaluation method.

A.3 More Ablation Studies

Reward Function To assess different accuracy rewards, we conduct experiments by training Qwen2.5-VL-7B-Instruct for the RL stage only, as shown in Table 5a. 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 both benchmarks. This finding underscores the importance of adjusting the reward function to the specific type of answers.

Image Diversity & Question Complexity Our ChartRQA dataset is characterized by two key features: diverse chart images and complex questions requiring step-by-step reasoning. To investigate the importance of these factors in RL training, we select samples from ChartRQA-RL that only include line, bar, and pie chart types, which are the same chart types found in the ChartQA dataset, and train the model using RL only. As shown in Table 5a, without increasing chart type diversity, the complex questions in ChartRQA still substantially enhance the model's reasoning ability. Furthermore, using the full ChartRQA dataset, which includes a wider variety of chart images, leads to further improvements on CharXiv-RQ.

SFT Data Composition 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 5b 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

RL Setting	ChartQA	CharXiv-RQ
Accuracy Reward		
ED	89.88	44.0
ED + SM	**90.28**	**45.2**
RL Training Set		
QA	89.32	42.1
QA + RQA†	**90.32**	44.6
QA + RQA	90.28	**45.2**

(a) Ablation study on reward and training set in RL.

SFT Setting	ChartQA	CharXiv-RQ
SFT Training Set		
RQA-SFT&RL + QA	88.40	41.2
RQA-SFT	**89.88**	**44.5**
SFT Dataset		
TinyChart	84.80	36.1
ChartGemma	86.72	39.1
ChartRQA-SFT	**90.20**	**45.0**

(b) Ablation study on training set and different datasets in SFT.

Table 5: Ablation studies within RL and SFT stages. ED and SM are the abbreviations of Edit Distance and Soft Matching. RQA† indicates that only samples with chart types of line, bar, and pie in ChartRQA are used for training.

the SFT stage, resulting in more rigid thinking processes and a significant loss of output diversity. Also, the direct-answer format of ChartQA discourages the model from developing the ability to break down problems into step-by-step thinking process.

Comparison with Existing Chart Datasets To enable a fair comparison between ChartRQA and existing chart SFT datasets, we replace the SFT dataset with TinyChart and ChartGemma, while keeping all other settings consistent. TinyChart is a comprehensive dataset that integrates multiple open-source datasets and comprises a variety of tasks. To ensure that the model focuses on chart understanding and reasoning, we exclude the Chart-to-text and Chart-to-table generation tasks. For the RL stage, we use a combination of ChartQA and ChartRQA-RL for 1 epoch of training. As shown in Table 5b, Chart-R1 trained on ChartRQA-SFT achieves the best performance on both benchmarks. The results indicate that the unified thinking and answer format and the effective step-by-step reasoning process in ChartRQA are key factors in enhancing the model's reasoning ability.

A.4 ChartRQA Analysis

We count the quantity and distribution of different chart types across the training and test sets of ChartRQA, as detailed in Table 6. The distribution among the various types to be well-balanced. Furthermore, Figures 7 and 8 provide visualization examples of 24 chart types from the ChartRQA dataset, showcasing both single-chart and multi-chart formats, respectively.

A.5 Prompts

To enhance transparency and reproducibility, we provide the exact prompts used for dataset generation and evaluation. For data generation, we employ Gemini-2.5-Flash to generate both plotting code and QA pairs for data construction. Figure 9 illustrates the prompt used for plotting 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. Figures 10 and 11 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 12 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.

A.6 The Use of AI Assistants

In this work, we used LLMs as writing tools to improve language clarity and readability. These models helped refine the text and enhance the presentation of our ideas. All research concepts, experiments, implementations, and analyses were conducted independently by the authors.

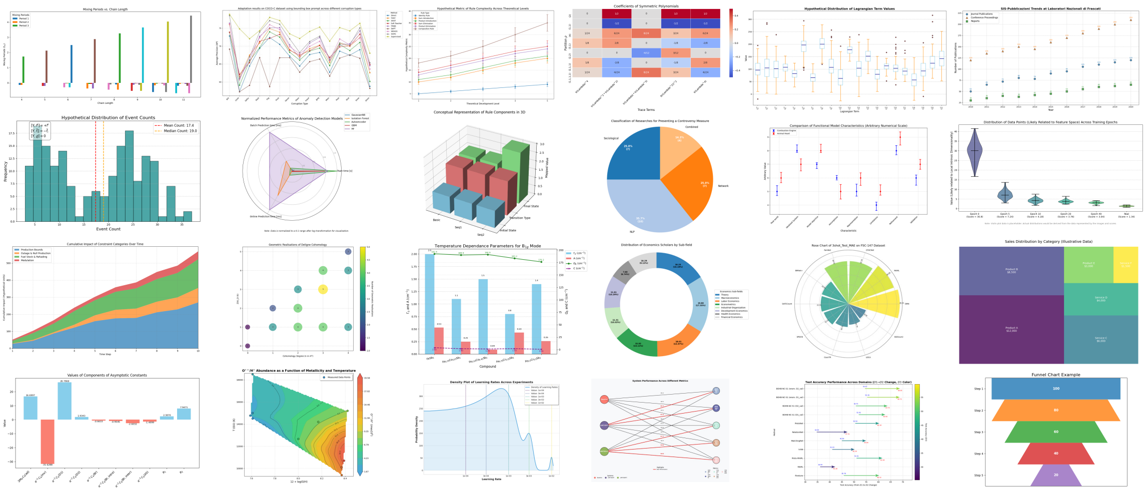

Figure 7: Single-chart samples of 24 chart types from ChartRQA.

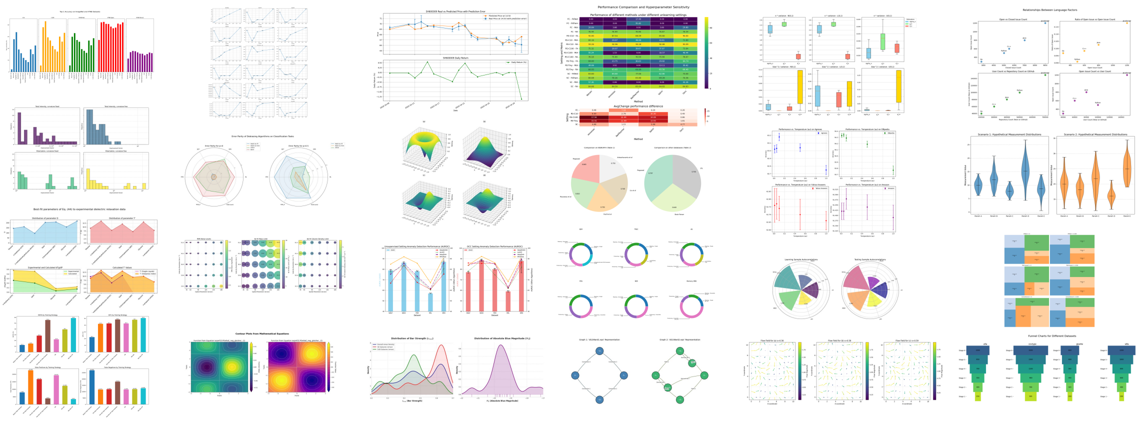

Figure 8: Multi-chart samples of 24 chart types from ChartRQA.

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

Table 6: 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.

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 9: 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 10: 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 11: 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., α 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 12: Prompt for ChartRQA evaluation using GPT-4o.