ChartPoint: Guiding MLLMs with Grounding Reflection for Chart Reasoning

Anonymous ICCV submission

Paper ID 4721

Abstract

Multimodal Large Language Models (MLLMs) have emerged as powerful tools for chart comprehension. However, they heavily rely on extracted content via OCR, which leads to numerical hallucinations when chart textual annotations are sparse. While existing methods focus on scaling instructions, they fail to address the fundamental challenge, i.e., reasoning with visual perception. In this paper, we identify a critical observation: MLLMs exhibit weak grounding in chart elements and proportional relationships, as evidenced by their inability to localize key positions to match their reasoning. To bridge this gap, we propose PointCoT, which integrates reflective interaction into chain-of-thought reasoning in charts. By prompting MLLMs to generate bounding boxes and re-render charts based on location annotations, we establish connections between textual reasoning steps and visual grounding regions. We further introduce an automated pipeline to construct `ChartPoint-SFT-62k`*, a dataset featuring 19.2K high-quality chart samples with step-by-step CoT, bounding box, and re-rendered visualizations. Leveraging this data, we develop two instruction-tuned models, ChartPoint$_{Q2}$ and ChartPoint$_{Q2.5}$, which outperform state-of-the-art across several chart benchmarks, e.g., +5.04% on ChartBench.*

1. Introduction

Recently, with Large Language Models (LLMs) demonstrating strong understanding and generalization capabilities [4, 8, 50, 55], Multimodal Large Language Models (MLLMs) have become the mainstream for processing multimedia data such as images and videos [5, 36, 41, 49]. Charts, as an intuitive way to present complex data, are widely adopted in documents and on the internet. However, current MLLMs heavily rely on optical character recognition (OCR) results when processing charts. When the text information extracted by OCR is limited, the MLLMs struggle to interpret the charts accurately, even leading to numerical hallucinations [35, 59, 65]. Thus, extracting chart content accurately and attaining profound chart comprehension continues to be challenging tasks.

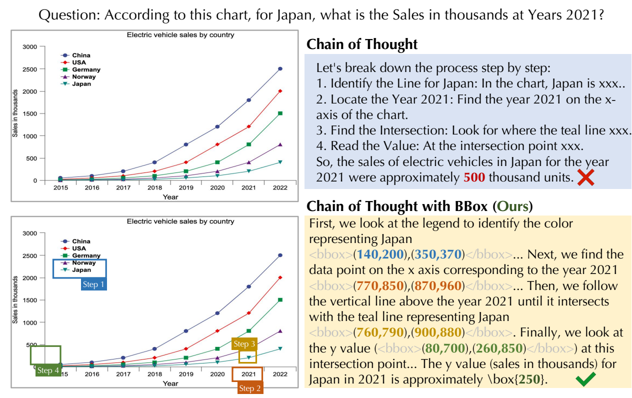


Figure 1. Comparison between vanilla CoT and proposed CoT with bounding box reflection on Qwen2-VL [58]. Vanilla CoT fails to introduce visual-level reflections. We re-render the generated BBox on the query chart to verify area focus and successfully improve the precision of the extracted numbers.

Existing methods attempt to address this issue through Supervised Fine-Tuning (SFT), including using more instruction-tuning data [21, 35, 43], increasing the chart resolution [75], or adopting more meticulously crafted alignment training techniques [44, 66, 68]. However, MLLMs still exhibit a limited perception of chart content. Recently, the inference-time scaling law and the reasoning models trained on it have exhibited impressive and in-depth reasoning capabilities [20, 82]. Chain-of-Thought (CoT) training has notably enhanced LLMs' proficiency in mathematics, logic, and code [60, 79]. This motivates us to refine reasoning paradigms and inference formats of MLLMs on charts, especially in scenarios with sparse text annotations.

Do current MLLMs truly grasp the correct logic for chart interpretation? As depicted in Fig. 1, while the MLLMs present reasonable steps for chart-reading, the numbers they extract still contain significant errors. This situation prompts a crucial question: do MLLMs rely excessively on the extracted numbers when interpreting charts, thus lacking the capacity to read from chart elements and proportional relationships? To explore this, we employ the MLLMs [6, 58] with satisfying localization capabilities, which can denote object positions using bounding boxes (BBox) or points. We prompt the model to point out the positions that match each reasoning step. Regrettably, MLLMs either overlook this request or generate entirely irrelevant positions. This implies that while the CoT approach bolsters the MLLM's logical processing based on numbers, it fails to enhance the model's fundamental numerical perception. Although CoT generates more inference tokens, it fails to enable additional interactions with chart or visual tokens, leading to limited perceptual improvement of MLLMs [27, 51]. Hence, we enhance CoT by incorporating a reflective interaction process, where the model outputs BBoxes and engages with re-rendered charts (Fig. 1). Hence, we construct CoT data with BBox reflection called PointCoT. We enhance the model's reasoning chain through a structured inference process and introduce an automated annotation pipeline leveraging chart-code pairs and advanced LLMs for precise step decomposition and key position localization.

This pipeline consists of four stages. 1) *Step Decomposition*: We collect high-quality chart-code pairs and use LLMs to generate a numerical question and corresponding CoT reasoning steps. The LLM labels each step as Grounding (requiring chart data extraction) or Reasoning. We will add point markers on the chart for all grounding steps. 2) *Code Editing*: LLMs modify the code for all grounding steps by inserting special characters at key positions for easier position extraction. Directly employing MLLMs is unreliable for this task. Hence, we employ LLM-based code editing to achieve high success. Thus, each grounding step has a corresponding edited code. 3) *Code Rendering*: We execute all modified code and re-render the charts. If any CoT step fails or triggers warnings, we discard the sample. 4) *Position Localization*: We perform OCR on each rendered chart to extract embedded character positions. Through format checks, we ultimately derive BBoxes for grounding steps. Ultimately, we construct 19.2K samples, each containing a detailed CoT process and position annotations. We further present `ChartPoint-SFT-62k`, a dataset of 62.3K instructions, along with two SFT models called ChartPoint$_{\text{Q2}}$ and ChartPoint$_{\text{Q2.5}}$. We achieve significant improvements across chart benchmarks, demonstrating the effectiveness of PointCoT. Our contributions are summarized as follows:

a) We introduce PointCoT, which enables the MLLM to verify whether its reasoning steps align with the chart content using generated bounding boxes.
b) We present `ChartPoint-SFT-62k`, a dataset containing 63.2K instruction-tuning samples. We also provide a data annotation pipeline to label the corresponding chart locations for CoT steps.
c) We propose the ChartPoint$_{\text{Q2}}$ and ChartPoint$_{\text{Q2.5}}$ based on proposed instruction data. Extensive experiments demonstrate that our models achieve state-of-the-art performance in chart understanding benchmarks.

2. Related Works

Multimodal Large Language Models adopt projectors to connect LLMs with visual encoders to understand images and demonstrate remarkable performance [83]. Some works employ QFormers [28] for modal alignment on large image-text pair datasets [2, 5, 28, 73]. Other works further simplify the architecture with a linear projector and extend the instruction tuning paradigm to visual tasks [36, 67]. Training strategies and data quality are crucial for the development of MLLMs. The GPT series [9, 48, 50, 82] and Claude series [3] are the models with SOTA performance. The LLaMA series [19, 54–56] initially leads the open-source community and spawns works like the LLaVA series [36–38]. The Qwen series [4–6, 58, 69, 70] and Intern series [10, 13–15, 17, 52, 77] have further elevated the performance of open-source models to SOTA level. The DeepSeek series [8, 16, 20, 31, 32, 41, 61] and Mistral series [24] conduct in-depth explorations of the Mixture of Experts architecture for MLLMs.

Chart Reasoning involves using MLLMs for tasks like question answering, description, analysis, and summarization of charts. Two-stage methods center on generating intermediate chart representations via specialized extraction modules. These representations can take forms such as markdown, as explored in [25, 33, 34], or dictionaries, as seen in [12, 62]. Subsequently, they are supplied as text prompts to LLMs. End-to-end methods attempt to optimize MLLMs with more chart-related instructions [21, 64]. Alignment training is employed to supplement prior knowledge in the chart domain, e.g., tabular [11, 35, 44], markdown [72, 74], JSON [68] or dictionaries [23]. ChartThinker [39] and DOMINO [57] propose the CoT for chart reasoning, and LaMenDa [84] further integrates step-by-step reasoning into the supervised fine-tuning stage. TinyChart [75] upsamples the chart resolution and achieves a notable performance improvement. Moreover, recent works [66, 71] attempt to combine the advantages of the above approaches using the mixture of experts architecture.

Multimodal Chain of Thought aims to extend text-based CoT reasoning [20, 60] to multimodal scenarios to enhance performance in tasks requiring logical reasoning. Some two-stage works either convert visual information into text [46, 47, 79] or sample key image information (e.g., region [51] or coordinate [26]). GoT [76] generates directed acyclic graphs to assist reasoning. Recently, structured reasoning is proposed to enhance the robustness of the CoT. Both InsightV [18] and LLaVA-CoT [63] propose a reasoning framework based on human design to solve a wide range of visual question-answering problems. Further research aims to enhance the interaction between the reasoning steps and the query image in structured scenarios [27, 53].

ICCV
#4721

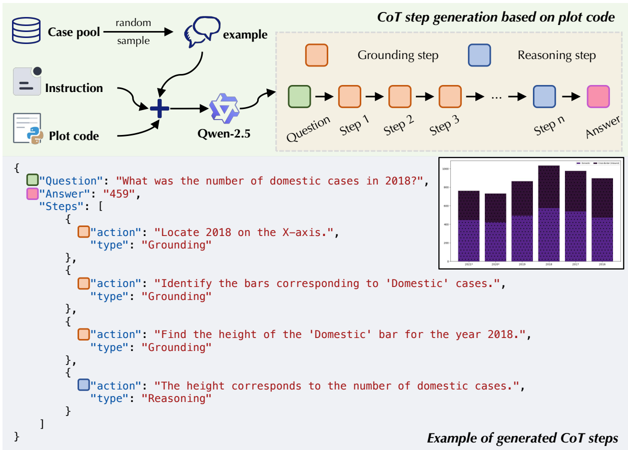


Figure 2. Chain of thought step generation based on plot code.

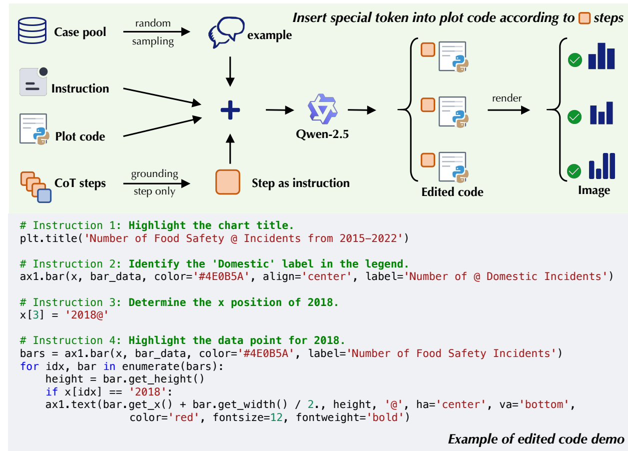


Figure 3. The pipeline of code editing with grounding steps.

3. Proposed Method

3.1. PointCoT

To enhance the reasoning process, we focus on constructing extensive thinking-chain data for chart-based Q&A while leveraging coordinate points to guide the model's attention to relevant chart regions. To ensure the model learns correct chart-reading logic, we select charts without datapoint annotations, preventing it from extracting answers directly via OCR. Specifically, our metadata construction is based on the ChartAlign dataset [66], which comprises one million quadruples (table, JSON, code, chart) sourced from ChartQA [42], PlotQA [45], and ChartY [12]. Our objective is to generate chain-of-thought reasoning data for charts and incorporate coordinate-based cues at each step to justify the model's focus region. The following sections detail the step-by-step process of constructing the training data.

3.2. Construction of Structured Reasoning

Researchers typically employ advanced LLMs to decompose and expand the reasoning process of the text data, aiming to obtain long chain-of-thought inference processes. Recent studies have also demonstrated that distillation learning based on such data enables smaller models to acquire strong reasoning abilities [20]. Unlike general visual Q&A tasks that require diverse knowledge and reasoning styles, chart Q&A exhibits a structured thought process, i.e., the model infers correct numbers from visual elements like legends and coordinate systems through consistent steps, which can be enhanced with structured reasoning training. Fig. 2 elaborately outlines the process of our reasoning data construction. Our primary focus lies in straightforward chart comprehension, centered around chart data points Q&A. Although the reasoning process appears structured, this structure does not arise from artificial constraints. Instead, it emerges naturally from the inherent logic of chart reading, imparting a degree of structural consistency to the decomposed CoT steps.

Fig. 2 presents an example chart and the generated JSON. First, we utilize the teacher model (i.e., Qwen2.5-72B [70]) to pose a datapoint-related question based on the plotting code. We require the teacher to provide a step-by-step reasoning process and the final answer. We employ few-shot examples to standardize the step-decomposition format and ask the teacher model to classify each sub-step into two categories: *Grounding* and *Reasoning*. Refer to Appendix B for the detailed prompt. Grounding steps focus on identifying the positions of chart elements, such as locating points on the axes or entries in the legend. Reasoning steps involve making logical inferences based on information obtained from previous grounding steps. This classification helps incorporate specific bounding boxes for steps that require element localization, thereby offering precise positional guidance. Finally, we instruct the teacher model to generate outputs in JSON format. Samples that pass both the format validation and key integrity checks proceed to the following processing stage.

3.3. Construction of Point Annotation

Our goal is to incorporate location supervision into all grounding steps, guiding the model to follow human-like chart-reading logic. We believe the generated bounding boxes not only validate the grounding steps but also encourage the model to re-examine the original input chart. Therefore, we implement point-based CoT training through grounding. MVoT [27] also achieves similar observations in other structured scenarios, e.g., puzzle-solving games.

Fig. 3 elaborately depicts how we add the position points to all grounding steps. Our modifications are based on revising the plotting code and OCR on the re-rendered chart. Specifically, we instruct the teacher model to identify the relevant elements (e.g., legend or title) or positions (e.g., datapoints or corresponding horizontal and vertical coordinates) for each grounding step. Next, the teacher model

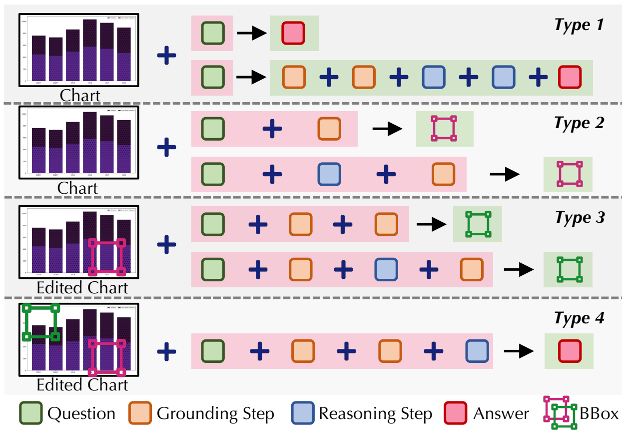


Figure 4. The process pipeline for constructing instruction data. The red / green indicates the instruction prompt / ground truth.

Table 1. Data processing steps and corresponding success rate. # indicates the number of instructions.

Processing Step	Meta	CoT	Code	Render	OCR	QA#
Chart Number	66.84K	64.28K	48.75K	24.88K	19.2K	62.3K
Success Rate	-	96.17%	75.84%	51.04%	77.17%	-

modifies the plotting code based on the identified positions by inserting a special symbol into the chart element text or marking a specific position using `plt.text()`. This insertion not only highlights key positions but also facilitates the quick detection of unique characters with OCR tools.

After passing the integrity check, the edited code is re-rendered to generate the updated chart. We then apply OCR to the re-rendered chart to extract the coordinates of the inserted special characters. To enhance extraction accuracy and success rates, we employ multiple OCR tools sequentially. A minimum width is defined for the bounding boxes generated by special characters, and any boxes more minor than the threshold are adjusted based on the center point and the pre-set width. Each grounding step is associated with an edited code, a re-rendered chart, and the detected positions from OCR. Refer to Appendix A for details.

3.4. Construction of Instruction

After obtaining the bounding boxes for all grounding steps, we begin constructing instruction data with location annotations. Fig 4 illustrates the process to construct `ChartPoint-SFT-62k`, which primarily includes four formats and 62K Q&A pairs.

Type 1: Standard VQA. The raw chart and question are used as input. 1) Supervised with ground truth answer. Unlike previous ChartQA [42], the data points are not directly labeled with text, making the questions more challenging. 2) Supervised with CoT steps and the answer as long text supervision. Here, the bounding boxes from the grounding step are excluded to prevent potential data leakage and avoid affecting other formats. *Type 2: Localization Task.* Different from direct Q&A, we introduce intermediate steps

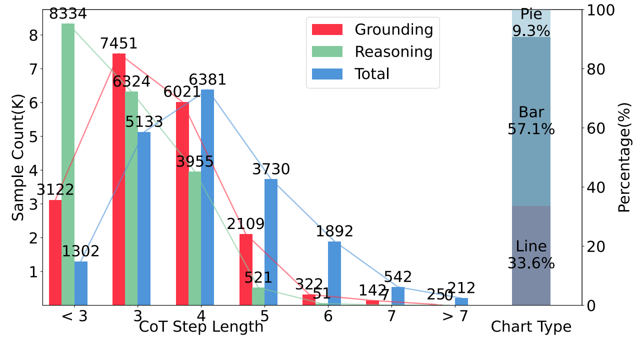


Figure 5. Statistic information of `ChartPoint-SFT-62k`. Left: Statistics on the number of CoT steps w.r.t. grounding, reasoning, and total steps. Right: chart type distribution.

Table 2. Instruction data used for ChartPoint superivised training.

Dataset	Description	Number
Chart Knowledge Alignment Stage		
MMC-Instruct [35]	VQA / Summariztion/ Reasoning	410K
ChartGemma [43]	VQA / Summariztion/ Reasoning	160K
ChartQA [42]	VQA	28K
ChartBench [65]	VQA	30K
Chart Specific Annealing Tuning Stage		
`ChartPoint-SFT-62k`	VQA / Reasoning	62K

into the query prompt. The ground truth is changed from the answer to the predicted bounding box, which is a localization task. *Type 3: Reasoning with Edited Chart.* The bounding box annotations in the previous grounding steps will be redrawn on the vanilla chart to attract attention to the key position, aiding the model in learning the correct visual reasoning logic. If the next step is also a grounding step, the model will continue to predict the next bounding box based on the edited chart. *Type 4: Reasoning Steps.* If the next step is the reasoning step, it will be added to the query prompt directly. Once the final step is processed, the supervised ground truth will be the final answer.

3.5. Quality Control

Considering the lengthy data generation process, we implement quality control at every step and track success rates. As shown in Tab. 1, we randomly sample 66.84k quadruples from ChartMoE-Align [66]. 1) We expand the reasoning process based on the plot code and perform the integrity check on the generated JSON (Fig. 2). We employ GPT-4o [49] to review the generated Q&A given the metatable data to filter out mismatched samples. The pass rate is 96.17%. 2) We modify the plotting code by incorporating the grounding step as the instruction (Fig. 3). We ensure the code integrity and verify the presence of the required unique character in the code. The pass rate is 75.84%. 3) We execute the modified code to render the edited charts. One case will be discarded if any code execution fails, resulting in a lower success rate of 51.04%. 4) We use OCR to detect special characters and extract the bounding boxes. We discard the cases where OCR fails or detects multiple occurrences.

ICCV #4721

Table 3. The relaxed accuracy (%) performance on ***ChartQA***. Ada.: Adaptive input resolution. Methods are sorted by relaxed average accuracy@0.05. All results are reproduced in the same inference manner by officially released model weights and prompts.

Models	Para.	Baseline	Res.	Relax Acc @0.05			Relax Acc @0.10			Relax Acc @0.20		
				Human	Aug.	Avg.	Human	Aug.	Avg.	Human	Aug.	Avg.
General MLLMs												
LLaVA-v1.5 [38]	13B	Vicuna [80]	@336	25.36	18.56	21.96	28.56	23.52	26.04	32.56	30.72	31.64
Qwen-VL [5]	9.6B	Qwen [4]	@448	40.48	79.76	60.12	43.20	82.56	62.88	47.52	85.76	66.64
Phi-3.5-Vision [1]	4.2B	Phi-3.5[1]	Ada.	60.08	83.52	71.80	64.00	85.92	74.96	68.16	89.36	78.76
InternlmXC-v2 [17]	8B	InternLM-v2 [10]	@490	62.72	81.28	72.00	66.72	84.08	75.40	70.80	86.56	78.68
InternVL-v2.5 [14]	8B	InternLM-v2.5 [10]	Ada.	65.44	86.48	75.96	67.36	86.88	77.12	68.80	87.44	78.12
DeepSeekVL2 [61]	27B	DeepSeek-v2 [31]	@384	65.52	87.76	76.64	67.52	88.08	77.80	69.60	88.96	79.28
Qwen2-VL [58]	7B	Qwen2 [69]	Ada.	72.08	**94.24**	83.16	75.76	**94.72**	85.24	78.24	**95.76**	87.00
Qwen2.5-VL [6]	7B	Qwen2.5 [70]	Ada.	**78.96**	93.76	**86.36**	**81.12**	94.16	**87.64**	**83.60**	94.72	**89.16**
Specialist Chart Models												
Matcha [34]	282M	Pix2Struct [25]	Ada.	37.12	86.64	61.88	39.84	87.36	63.60	43.52	88.56	66.04
ChartVLM [62]	13B	Vicuna [80]	Ada.	42.08	82.48	62.28	43.84	82.88	63.36	46.00	83.28	64.64
DocOwl-v1.5 [22]	8B	mPLUG-Owl2 [74]	@448	47.44	91.52	69.48	51.92	92.08	72.00	56.72	93.12	74.92
Deplot [33]	13.2B	LLaVA-v1.6 [37]	Ada.	53.44	87.68	70.56	56.80	88.48	72.64	60.64	90.08	75.36
OneChart [12]	13.3B	LLaVA-v1.6 [37]	@1024	54.48	87.12	70.80	57.60	87.84	72.72	62.00	88.64	75.32
ChartLlama [21]	13B	LLaVA-v1.5 [38]	@336	58.40	93.12	75.76	61.20	93.60	77.40	63.52	94.00	78.76
ChartGemma+PoT [43]	3B	PaliGemma [7]	@448	67.84	85.28	76.56	68.64	85.84	77.24	69.84	86.32	78.08
ChartAst [44]	13B	Sphinx [30]	@448	64.88	93.12	79.00	66.24	93.84	80.04	67.44	94.32	80.88
TinyChart+PoT [75]	3B	TinyLlava [78]	@768	70.24	90.72	80.48	71.20	91.44	81.32	72.40	92.56	82.48
ChartMoE+PoT [66]	8B	InternlmXC-v2 [17]	@490	78.32	90.96	84.64	80.16	92.32	86.24	82.08	93.60	87.84
ChartPoint$_{Q2}$	7B	Qwen2-VL [58]	Ada.	76.12	**94.48**	85.28	78.36	94.96	86.66	81.28	95.12	88.20
ChartPoint$_{Q2.5}$	7B	Qwen2.5-VL [6]	Ada.	**81.36**	94.12	**87.74**	**82.40**	**95.24**	**88.82**	**84.48**	**95.76**	**90.12**

This step achieves a success rate of 77.17%. Finally, we construct 19.2K charts and 62.3K instruction data as illustrated in Fig. 4. We randomly sample 100 cases, which are reviewed by at least three experts to evaluate the bounding box quality of the grounding step based on the process in Fig. 2. 91% of the cases meet the desired standard.

3.6. Statistics

Fig. 5 presents the statistics of `ChartPoint-SFT-62k`. As shown in Fig. 5 (left), we carefully count all the CoT steps and organize the samples based on the length of the CoT steps. Most samples contain 3-5 CoT steps. Notably, the grounding steps are typically longer (length $>$ 3) than the reasoning steps, which are predominantly short (length $\leq$ 3) and generally focus on summary-style analyses. This is because our Q&A primarily addresses numerical data points without requiring complex numerical reasoning, allowing the dataset to effectively capture the essential *visual logical* based more on grounding. As shown in Fig. 5 (right), we primarily focus on three chart types, i.e., line (33.6%), pie (9.3%), and bar (57.1%) charts, which is consistent with the distribution of mainstream chart datasets [42, 45].

3.7. ChartPoint

We integrate bounding box reflection into the inference. The baseline's grounding ability is critical for instruction tuning. Hence, we select Qwen2-VL [58] and Qwen2.5-VL [6] as baselines due to their comprehensive grounding capabilities. They can be deployed based on LLaMA-Factory [81] to conduct convenient training. We perform a two-stage full fine-tuning process using the data in Tab. 2. We utilize high-quality instruction data (including real-world annotated and diversely synthesized charts) for chart knowledge alignment to enhance the baseline's performance. Then, we refresh the learning rate and conduct chart-specific annealing tuning in our PointCoT manner. The SFT models are named ChartPoint$_{Q2}$ and ChartPoint$_{Q2.5}$, respectively.

4. Experiment

4.1. Implement Details

ChartPoint is initialized from Qwen [6, 58], which employs a dynamic resolution input strategy. We keep all numerical coordinates within the range of $0-999$ to adapt to the tokenizer and the pretrain format of the coordinate system. We use LLaMA-Factory [81] for supervised fine-tuning over 2 epochs. In the first 1% of the training steps, we implement a warmup phase with a learning rate of $5e-5$. We adopt the AdamW [40] optimizer with a constant weight decay of 0.1 throughout the training. The gradient clip is set to 1.0. We conduct gradient accumulation with an equivalent batch size of 64 and train using *bfloat16* precision. The training process consumes around 262 GPU Hours (A100-40G).

4.2. Benchmarks

ChartQA [42] test split comprises $1,250$ questions from both human-generated and augmented segments. The charts are sourced from web crawls with three prevalent chart types. ChartQA requires the model to respond to questions with only a single word or phrase and employs a lenient matching method to verify the correctness of the answers. Considering the impact of inference length on performance,

ICCV
#4721

Table 4. The accuracy (%) performance on ***ChartBench***. Our proposed ChartPoint consistently outperforms other MLLMs remarkably.

Models	Regular Type				Extra Type							ALL
	Line	Bar	Pie	Avg.	Area	Box	Radar	Scatter	Node	Combin.	Avg.	
General MLLMs												
LLaVA-v1.5 [38]	29.12	21.26	17.28	22.10	21.73	20.94	27.50	23.47	36.80	24.30	24.96	23.38
Qwen-VL [5]	38.00	20.71	38.24	29.46	28.83	24.17	35.00	19.50	18.50	25.50	26.56	28.18
Mini-Gemini [29]	34.88	36.12	40.40	36.77	31.20	23.33	30.60	35.20	43.60	27.90	30.61	34.37
InternlmXC-v2 [17]	68.16	48.74	56.60	54.50	27.47	25.33	40.10	52.93	50.40	46.20	39.72	48.41
InternVL-v2.5 [14]	75.20	48.31	52.00	55.09	32.00	20.00	44.00	45.33	70.00	48.00	42.11	49.43
DeepSeekVL2 [61]	69.28	49.66	47.40	53.71	40.80	44.40	40.50	76.14	45.40	59.50	51.31	53.02
Qwen2-VL [58]	74.40	50.77	63.00	58.36	**56.93**	40.00	50.00	**81.33**	64.00	68.00	**59.40**	58.90
Qwen2.5-VL [6]	**80.88**	**54.06**	**68.20**	**62.73**	37.33	**46.13**	**51.90**	72.27	**74.40**	**74.00**	57.26	**60.91**
Specialist Chart Models												
Matcha [34]	6.80	5.05	3.60	5.18	0.27	1.60	6.20	3.46	5.40	4.80	5.81	4.84
ChartVLM [62]	21.92	14.16	10.50	15.16	7.47	7.87	8.00	7.87	5.40	10.50	8.38	11.96
ChartLlama [21]	26.80	18.83	20.80	20.99	14.27	12.00	24.30	27.73	26.20	25.80	21.71	21.31
TinyChart [75]	32.40	25.81	22.50	26.71	10.13	14.80	13.40	28.14	10.80	21.60	22.56	22.51
Deplot [33]	31.20	26.46	24.00	27.09	21.34	13.34	24.00	41.34	42.00	31.00	31.57	27.62
OneChart [12]	41.28	30.28	29.60	32.65	19.07	13.20	24.60	38.53	34.80	27.90	31.91	29.93
DocOwl-v1.5 [22]	49.60	31.69	31.54	35.68	12.27	23.33	22.50	36.13	29.60	38.80	27.38	32.05
ChartGemma [43]	50.48	38.21	32.10	39.89	28.27	24.13	28.10	48.00	41.80	43.40	42.47	38.46
ChartMoE [66]	71.44	51.57	52.80	56.31	38.40	24.13	40.20	62.67	58.00	49.20	55.58	51.67
ChartPoint$_{Q2}$	79.84	54.58	68.24	63.04	**58.20**	44.12	52.40	**83.67**	68.24	68.92	62.09	62.61
ChartPoint$_{Q2.5}$	**82.40**	**58.88**	**71.40**	**66.71**	51.44	**48.33**	**56.90**	77.27	**78.00**	**80.20**	**65.03**	**65.95**

instead of prompting the model to produce the shortest possible answers, we adopt a template-based answer extraction method, i.e., *provide your final answer in* $\backslash box\{\}$. Refer to Appendix B for details. This approach effectively enhances the performance of mainstream models.

ChartBench [65] offers charts that lack data point annotations. It encompasses 9 main categories and 42 subcategories, with each sub-category housing 50 charts. ChartBench emphasizes the reliability of chart numbers, presenting a stiffer challenge since models are unable to obtain precise answers via OCR. The models must understand each element of the chart to estimate values close to the ground truth. This benchmark uses a relaxed accuracy similar to ChartQA, and we also adopt the inference prompt of template extraction to boost model performance.

4.3. Comparative Models

We divide all methods into two groups: general MLLMs and those specifically designed for chart understanding.

General MLLMs. We compare LLaVA-v1.5 [38], which paved the way for image-text interaction through visual instruction fine-tuning. We also compare the QwenVL series, including v1 [5], v2 [58], and v2.5 [6]. Due to its strong base performance, we set this series as the baseline for our ChartPoint. We select Phi-3.5-Vision [1], which is easy to deploy on the edge devices, and the Intern series for their high performance, such as InternlmXComposer-v2 [17] and InternVL-v2.5 [14]. We also provide the result of DeepSeekVL2 [31], which is based on the MoE architecture. Note that we chose the versions of these models at around 10B for fair comparisons.

Specialist chart models. We provide classic chart methods like Matcha [34] and Deplot [33]. However, we adopt LLaVA-v1.6 [37] to further analyze and summarize their output for meaningful comparisons. We also compare ChartVLM [62], ChartAst [44], DocOwl-v1.5 [22], OneChart [12], and ChartLLama [21], which are fine-tuned with chart-specific instructions. Since the Program of thought (PoT) can effectively improve the numerical calculation ability of MLLMs, we select ChartGemma [43], TinyChart [75], and ChartMoE [66] for comparisons.

4.4. Comparison with SOTA

Comparisons on ChartQA. Tab. 3 presents the performance of ChartPoint on ChartQA. We report the relaxed accuracy for three different margins and provide detailed results for two distinct parts. ChartPoint significantly outperforms the baselines, e.g., ChartPoint$_{Q2}$ 83.16% [58] vs. 85.28% (+2.12%↑) and ChartPoint$_{Q2.5}$ 86.36% [6] vs. 87.74% (+1.38%↑). Even though the Qwen-VL series models demonstrate sufficiently high baseline performance, ChartPoint still manages to achieve remarkable enhancements, especially in the challenging Human-annotated part. This indicates that point-based CoT training can significantly improve the model's ability to read and understand charts. Notably, ChartPoint also outperforms PoT-based methods [43, 66, 75]. For example, when compared with ChartMoE+PoT [66], ChartPoint attains 84.64% vs. 87.74% (+3.10%↑). This implies that increasing the reasoning length contributes to enhancing the model's numerical and logical capabilities, effectively overcoming scenarios involving extensive numerical calculations.

Comparisons on ChartBench. Tab. 4 shows the performance of ChartPoint on ChartBench, where we report the detailed performance across 9 types of charts. Compared to ChartQA, ChartPoint demonstrates more significant im-

Table 5. Ablation study of training data in Tab. 2. CoT: stage 2 adopts the CoT data generated by Fig. 2. PointCoT: stage 2 adopts `ChartPoint-SFT-62k`.

Settings	ChartQA			ChartBench		
	Human	Aug.	Avg.	Regular	Extra	Avg.
Qwen2-VL	72.08	94.24	83.16	58.36	59.40	58.90
+Stage1	72.76	94.72	83.74	60.62	60.12	60.39
+Stage1+CoT	73.58	94.64	84.11	60.94	60.54	60.76
+Stage1+PointCoT	76.12	94.48	85.30	63.04	62.09	62.61
Qwen2.5-VL	78.96	93.80	86.38	62.73	58.93	61.67
+Stage1	79.16	93.88	86.52	64.22	60.82	62.68
+Stage1+CoT	79.76	93.52	86.64	64.48	61.16	62.98
+Stage1+PointCoT	81.36	94.12	87.74	66.71	65.03	65.95

Table 6. Ablation study on different MLLMs. We report the average relax accuracy@0.05 on ChartQA and ChartBench. PointCoT: stage 2 adopts `ChartPoint-SFT-62k`.

Model	ChartQA	Δ	ChartBench	Δ
Qwen-VL [5]	65.70	-	28.18	-
+PointCoT	66.12	+0.42	27.92	-0.26
ChartMoE [66]	81.20	-	51.67	-
+PointCoT	81.36	+0.16	51.94	+0.27
Qwen2-VL [58]	83.16	-	58.90	-
+PointCoT	84.84	+1.68	62.12	+3.22
Qwen2.5-VL [6]	86.36	-	61.67	-
+PointCoT	87.48	+1.12	65.66	+3.99

provements on ChartBench, e.g., ChartPoint$_{Q2}$ 58.90% [58] vs. 62.61% (+3.71%↑) and ChartPoint$_{Q2.5}$ 60.91% [6] vs. 65.95% (+5.04%↑). While better OCR capabilities can enhance model performance on ChartQA, ChartBench focuses on data points without text annotations, which benefits more from superior chart element localization and reasoning abilities. This supports the advantage of point-based CoT over text-only CoT. Specifically, the improvement is more significant on *extra* type charts, e.g., ChartPoint$_{Q2.5}$ 57.26% [6] vs. 65.03% (+7.77%↑). This suggests that Point-based CoT training enables the model to develop a logical chart-reading process and comprehension skills, enhancing its generalization even to uncommon chart types.

5. In-depth Analysis

5.1. Ablation on Training Recipe

Tab. 5 presents the ablation study on our training recipe. As shown in Tab. 2, we conduct the high-quality chart knowledge alignment before instruction tuning (+Stage1). We design detailed reasoning steps based on advanced LLMs (Fig. 2) to ensure even smaller models (∼7B) can also benefit from inference scaling laws (+CoT). Additionally, we integrate grounding supervision into the CoT steps, enabling the model to continuously reflect on its reasoning and interact with input charts to refine the reasoning chain (+PointCoT). Since the baseline model is optimized for ChartQA during pre-training, the Stage1 alignment training yields marginal performance improvements (e.g., Qwen2-VL +0.58%↑, Qwen2.5-VL +0.14%↑). Direct distillation

Table 7. Ablation study of bounding box format on ChartQA. In the ground truth, we normalize the point number into 0-1 (retain 3/4 decimal) or 0-999 to indicate the grounding area.

Settings	Normalize	Decimal	Human	Δ	Aug.	Δ	ALL	Δ
Qwen2-VL	-	-	72.08	-	94.24	-	83.16	-
Type A	[0-1]	4	73.52	+1.44	93.84	-0.40	83.68	+0.52
Type B	[0-1]	3	74.68	+2.60	94.16	-0.08	84.42	+1.26
Type C	[0-999]	0	75.36	+3.28	94.32	+0.08	84.84	+1.68

Table 8. Ablation study of prompt engineering (PE) on ChartQA. Direct: PE from ChartQA. Match: inference step by step and extract final answer via designed pattern.

Model	PE	Human	Δ	Aug.	Δ	ALL	Δ
Qwen2-VL	direct	72.08	-	94.24	-	83.16	-
	match	73.84	+1.76	94.32	+0.08	84.08	+0.92
ChartPoint$_{Q2}$	direct	75.22	-	94.24	-	84.73	-
	match	76.12	+0.90	94.48	+0.24	85.28	+0.55

from reasoning steps also shows limited improvement because: 1) In Fig. 2, we adopt the LLM (not MLLM), so the reasoning process does not leverage chart information; 2) both ChartQA and ChartBench focus more on data point accuracy rather than numerical calculation or reasoning. Hence, textual CoT does not improve the model's accuracy in reading basic numbers from the chart. With grounding supervision, the model performance gets significantly improved, particularly on sparse-annotated ChartBench (Qwen2-VL +3.71%↑, Qwen2.5-VL +4.28%↑).

5.2. Ablation on Backbone

To demonstrate the effect of MLLMs for SFT based on PointCoT, we select two baseline models with relatively poor localization but strong chart-processing abilities for comparisons. As shown in Tab. 6, PointCoT is highly dependent on the underlying localization capabilities. Although both Qwen-VL [5] and ChartMoE [66] perform excellently in handling chart data, the reflection based on BBox fails to enhance their performance further. In contrast, both Qwen2-VL [58] and Qwen2.5-VL [6] can accurately indicate the objects using either points or BBoxes. Correspondingly, this enables PointCoT to work effectively, achieving a performance improvement of more than 1%.

5.3. Ablation on Bounding Box Format

Our proposed ChartPoint reflects on the chart regions by outputting $(X_{\text{top left}}, Y_{\text{top left}}), (X_{\text{bottom right}}, Y_{\text{bottom right}})$ as bounding boxes. Our observations reveal that the numerical representation format significantly impacts the tuning process. Table 7 presents three formats using baselines trained on `ChartPoint-SFT-62k` for one epoch without additional data or tricks. *Type A* normalizes numbers to four-decimal values between 0 and 1, representing relative positions on the chart. However, it yields only a marginal performance improvement of 0.52%. *Type B* rounds values to three decimal places. With the same data size and training

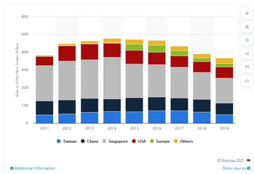
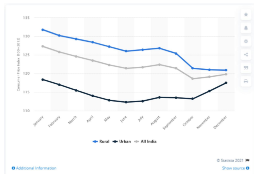
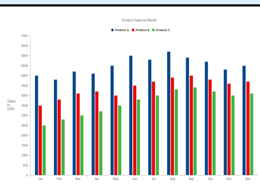
	Qwen2.5-VL-72B	GPT-4O	Ours
 **Question**: What was Asus' revenue in the United States in 2019? **Label**: 62.16	### Initial Assessment: The chart is showing Asus' revenue breakdown by region across the years from 2011 to 2019. The vertical axis represents "Sales in billion New Taiwan dollars". The x-axis shows the years. A legend is present at the bottom to represent different regions by colors. ### Step-by-Step Reasoning: 1. Identify the relevant data: From the legend, the USA is represented by the red-colored segment of each bar. 2. Locate the 2019 bar: The last bar on the right represents the year 2019. 3. Focus on the USA portion in 2019: The red portion of the stacked bar for 2019 is situated near the top of the stack. 4. Determine the height and value: The USA revenue is approximately 110. ### Final Answer: \box{110} ✗	### Initial Assessment: The chart is a stacked bar chart, which shows the sales data in billions of Taiwanese New Dollars (TWD) from 2011 to 2019. ### Step-by-Step Reasoning: 1. Identify the relevant data: The red represents USA. 2. Focus on the 2019 data: Look at the bar corresponding to the year 2019. 3. Find the USA's contribution: The height of the red section represents the sales in the United States. 4. Estimate the value: By looking at the chart, we can visually assess the size of the red section in 2019. It appears to be approximately around 120 billion TWD. ### Final Answer: \box{120} ✗	### Initial Assessment: The x-axis represents the years from 2011 to 2019. The y-axis represents the revenue in thousands. ### Step-by-Step Reasoning: 1. Locate the 2019 data: We locate the bar for the year 2019 on the x-axis(830,310),(900,730). 2. Focus on the USA data: We identify the segment of the bar that corresponds to the USA according to the legend(820,360),(920,430). 3. Estimate the value: We can see the height of the red segment for the bar in 2019(50,320),(130,500). The revenue value for Asus in the USA in 2019 is 60. ### Final Answer: \box{60} ✓
 **Question**: What was the Consumer Price Index for pulses and its products across rural areas in India in December 2018? **Label**: 117.5	### Initial Assessment: The chart shows the Consumer Price Index (CPI) for pulses and its products in India for rural, urban, and all India areas across the months from January to December. The y-axis represents the PI, and the x-axis represents the months. There is a legend at the bottom indicating that the blue line represents rural areas, the black line represents urban areas, and the gray line represents all India. ### Step-by-Step Reasoning: 1. Identify the relevant line: We focus on the blue line. 2. Locate the correct point in time: December is the last point on the x-axis. 3. Determine the corresponding value: We observe that it lies at approximately 121 on the y-axis. ### Final Answer: \box{121} ✓	### Initial Assessment: The chart shows the CPI changes over months from January to December 2018. The vertical axis represents the Consumer Price Index with a base year of 2012. The horizontal axis shows the months from January to December. ### Step-by-Step Reasoning: 1. Identify the relevant data: We look for the blue line, which represents the CPI for rural areas. 2. Focus on the December data: We locate December on the horizontal axis. 3. Estimate the value: We identify the corresponding point on the blue line for December and check its value on the vertical axis. ### Final Answer: \box{121} ✓	### Initial Assessment: The y-axis represents the Consumer Price Index. The x-axis shows the months from Jan. to Dec.. There is a legend that differentiates between Rural, Urban, and All India data. We are interested in the Rural data for Dec. ### Step-by-Step Reasoning: 1. Locate the data: We locate the "December" label on the x-axis(820,630),(890,720). 2. Focus on the blue data: We find the blue line represents the Rural data according to legend(350,750),(420,820). 3. Estimate the value: We trace the blue line to the point corresponding to December(850,360),(900, 400). The value on the y-axis at this point is 121. ### Final Answer: \box{121} ✓
 **Question**: According to this chart, what is the Sales of Product B at Month Oct? **Label**: 4800	### Initial Assessment: The chart illustrates the sales data for three distinct products, A, B, and C across different months from January to December. The vertical axis represents the sales in USD, and the horizontal axis indicates the months of the year. A legend at the top identifies the products: Product A is represented by navy blue, Product B by red, and Product C by green. ### Step-by-Step Reasoning: 1. Identify the Month: We look at the horizontal axis to locate the month of October. 2. Identify the Product: According to the legend, Product B is represented by red. 3. Read the Sales Value: The sales for Product B in October are slightly above 4700. ### Final Answer: \box{4700} ✓	### Initial Assessment: The chart is a bar chart titled "Product Sales by Month". The x-axis represents months from January to December. The y-axis represents sales in USD. ### Step-by-Step Reasoning: 1. Identify the relevant data: We look at the x-axis to locate the month of October.2. Find the October data: We focus on the red bar for October, which signifies Product B's sales. 3. Estimate the value: We check the height of the red bar against the y-axis to determine the sales value. The red bar for Product B in October reaches 4000 on the y-axis. ### Final Answer: \box{4000} ✗	### Initial Assessment: The x-axis represents the months from Jan to Dec, and the y-axis represents "Sales in USD". The legend indicates that the red bars represent Product B. ### Step-by-Step Reasoning: 1. Locate the data: We locate the month of Oct. on the x-axis (770,870),(830,930). 2. Focus on the red data: We find the red bar corresponding to Oct., which represents Product B(790,340),(820,920). 3. Estimate the value: Observing the height of the red bar for Oct.(40,320),(130,450), we can see it corresponds to a value of 4800 on the y-axis. ### Final Answer: \box{4800} ✓

Figure 6. Comparsion between Qwen2.5-VL-72B [6], GPT-4O [49] and ChartPoint$_{Q2.5}$ (ours). All models adhere to the output format required by the prompt. However, both Qwen2.5-VL and GPT-4O ignore the BBox instruction. With the reflective output of the BBox, our ChartPoint$_{Q2.5}$ has extracted precise numbers, and the BBoxes have provided sound explanations.

time, it achieves a 1.26% improvement, significantly outperforming type A. Further analysis suggests that Qwen's tokenizer splits decimals into three-digit segments, potentially increasing token-level training difficulty for Type A. *Type C* retains the baseline positioning format, which varies across MLLMs, using numbers between 0 and 999 to represent relative positions. This approach proves particularly beneficial for grounding training, leading to a 1.68% performance improvement in just one epoch. These findings highlight the importance of numerical representation in optimizing model performance.

5.4. Ablation on Prompt Engineering

To effectively utilize rule-based metrics for evaluation, researchers require models to respond with a direct number or phrase, i.e., *direct* prompt. However, we observe that for models with excellent instruction-following capabilities, performance can be further improved by extending the reasoning length. This conclusion is well-established in reasoning models [20, 82]. Still, it also applies to MLLMs that are not explicitly designed for reasoning, particularly when compared to prompts that generate only a single word. Tab. 8 illustrates two types of PE on both the baseline and our ChartPoint$_{Q2}$, with modifications applied exclusively to the reasoning prompt while keeping the model parameters unchanged. For Qwen2-VL, adjusting the PE results in a 0.92% performance improvement, particularly on the more challenging Human subset. Although ChartPoint$_{Q2}$ already demonstrated strong performance, the PE provides an additional 0.55% gain on ChartQA.

5.5. Case Visualization

Fig. 6 demonstrates specific cases from ChartQA and ChartBench. We choose the powerful Qwen2.5-VL-72B [6] and GPT-4O [49] for comparison with our ChartPoint$_{Q2.5}$. We request the models to output BBox when generating the CoT steps to support their reasoning (Appendix B). As shown in Fig. 6, only our ChartPoint$_{Q2.5}$ provide the BBoxes as required by the prompt, yielding more accurate numbers on charts with sparse text annotations.

6. Conclusion

We propose PointCoT, a multimodal CoT training method for chart understanding. We adopt the generated bounding boxes to verify whether the chain-of-thought reasoning steps are in line with the chart content. Specifically, we propose an automated annotation pipeline to provide the corresponding bounding boxes in the grounding steps and thus construct an instruction dataset. We provide two supervised fine-tuning models based on PointCoT data and conduct extensive experiments to demonstrate their effectiveness.

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint:2404.14219*, 2024. 5, 6

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, et al. Flamingo: A visual language model for few-shot learning. In *proceedings of NeurIPS*, pages 23716–23736, 2022. 2

[3] Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024. Anthropic Research Blog. 2

[4] Jinze Bai, Shuai Bai, Yunfei Chu, et al. Qwen technical report. *arXiv preprint:2309.16609*, 2023. 1, 2, 5

[5] Jinze Bai, Shuai Bai, Shusheng Yang, et al. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint:2308.12966*, 2023. 1, 2, 5, 6, 7

[6] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, et al. Qwen2.5-vl technical report. *arXiv preprint:2502.13923*, 2025. 1, 2, 5, 6, 7, 8

[7] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint:2407.07726*, 2024. 5

[8] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint:2401.02954*, 2024. 1, 2

[9] Tom Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. In *proceedings of NeurIPS*, pages 1877–1901, 2020. 2

[10] Zheng Cai, Maosong Cao, et al. Internlm2 technical report. *arXiv preprint:2403.17297*, 2024. 2, 5

[11] Victor Carbune, Hassan Mansoor, Fangyu Liu, et al. Chart-based reasoning: Transferring capabilities from llms to vlms. In *proceedings of NAACL*, 2024. 2

[12] Jinyue Chen, Lingyu Kong, Haoran Wei, Chenglong Liu, et al. Onechart: Purify the chart structural extraction via one auxiliary token. In *proceedings of ACMMM*, pages 147–155, 2024. 2, 3, 5, 6

[13] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint:2312.14238*, 2023. 2

[14] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint:2412.05271*, 2024. 5, 6

[15] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint:2404.16821*, 2024. 2

[16] Damai Dai, Chengqi Deng, Chenggang Zhao, RX Xu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint:2401.06066*, 2024. 2

[17] Xiaoyi Dong, Pan Zhang, Yuhang Zang, et al. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint:2401.16420*, 2024. 2, 5, 6

[18] Yuhao Dong, Zuyan Liu, Hai-Long Sun, Jingkang Yang, et al. Insight-v: Exploring long-chain visual reasoning with multimodal large language models. *arXiv preprint:2411.14432*, 2024. 2

[19] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The llama 3 herd of models. *arXiv preprint:2407.21783*, 2024. 2

[20] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint:2501.12948*, 2025. 1, 2, 3, 8

[21] Yucheng Han, Chi Zhang, Xin Chen, et al. Chartllama: A multimodal llm for chart understanding and generation. *arXiv preprint:2311.16483*, 2023. 1, 2, 5, 6

[22] Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, et al. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. In *proceedings of EMNLP*, pages 3096–3120, 2024. 5, 6

[23] Muye Huang, Lingling Zhang, Lai Han, Wenjun Wu, et al. Vprochart: Answering chart question through visual perception alignment agent and programmatic solution reasoning. *arXiv preprint:2409.01667*, 2024. 2

[24] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, et al. Mistral 7b. *arXiv preprint:2310.06825*, 2023. 2

[25] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, et al. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *proceedings of ICML*, pages 18893–18912, 2023. 2, 5

[26] Xuanyu Lei, Zonghan Yang, Xinrui Chen, Peng Li, and Yang Liu. Scaffolding coordinates to promote vision-language coordination in large multi-modal models. In *proceedings of COLING*, pages 2886–2903, 2025. 2

[27] Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, et al. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint:2501.07542*, 2025. 2, 3

[28] Junnan Li, Dongxu Li, Silvio Savarese, et al. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *proceedings of ICML*, pages 19730–19742, 2023. 2

[29] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, et al. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint:2403.18814*, 2024. 6

[30] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint:2311.07575*, 2023. 5

[31] Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint:2405.04434*, 2024. 2, 5, 6

[32] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, et al. Deepseek-v3 technical report. *arXiv preprint:2412.19437*, 2024. 2

[33] Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, et al. Deplot: One-shot visual language

reasoning by plot-to-table translation. In *Findings of ACL*, pages 10381–10399, 2023. 2, 5, 6

[34] Fangyu Liu, Francesco Piccinno, Syrine Krichene, et al. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering. In *proceedings of ACL*, pages 12756–12770, 2023. 2, 5, 6

[35] Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. MMC: advancing multimodal chart understanding with large-scale instruction tuning. In *proceedings of ACL*, 2023. 1, 2, 4

[36] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *proceedings of NeurIPS*, 2023. 1, 2

[37] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, et al. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 5, 6

[38] Haotian Liu, Chunyuan Li, Yuheng Li, et al. Improved baselines with visual instruction tuning. In *proceedings of CVPR*, 2024. 2, 5, 6

[39] Mengsha Liu, Daoyuan Chen, Yaliang Li, Guian Fang, and Ying Shen. Chartthinker: A contextual chain-of-thought approach to optimized chart summarization. In *proceedings of LREC-COLING*, pages 3057–3074, 2024. 2

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *proceedings of ICLR*, 2019. 5

[41] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, et al. Deepseek-vl: Towards real-world vision-language understanding. *arXiv preprint:2403.05525*, 2024. 1, 2

[42] Ahmed Masry, Do Xuan Long, Jia Qing Tan, et al. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *proceedings of ACL*, 2022. 3, 4, 5

[43] Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. Chartgemma: Visual instruction-tuning for chart reasoning in the wild. *arXiv preprint:2407.04172*, 2024. 1, 4, 5, 6

[44] Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, et al. Chartassistant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. In *proceedings of ACL*, pages 7775–7803, 2024. 1, 2, 5, 6

[45] Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. Plotqa: Reasoning over scientific plots. In *proceedings of CVPR*, pages 1527–1536, 2020. 3, 5

[46] Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *proceedings of CVPR*, pages 14420–14431, 2024. 2

[47] Debjyoti Mondal, Suraj Modi, Subhadarshi Panda, Rituraj Singh, and Godawari Sudhakar Rao. Kam-cot: Knowledge augmented multimodal chain-of-thoughts reasoning. In *proceedings of the AAAI*, pages 18798–18806, 2024. 2

[48] OpenAI. Gpt-4 technical report. *arXiv preprint:2303.08774*, 2023. 2

[49] OpenAI. Gpt-4o: A multimodal large language model. https://openai.com, 2024. Accessed: 2024-09-17. 1, 4, 8

[50] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training, 2018. 1, 2

[51] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, et al. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *proceedings of NeurIPS*, pages 8612–8642, 2025. 2

[52] Jing Shao, Siyu Chen, Yangguang Li, Kun Wang, et al. Intern: A new learning paradigm towards general vision. *arXiv preprint:2111.08687*, 2021. 2

[53] Xuan Shen, Yizhou Wang, Xiangxi Shi, Yanzhi Wang, et al. Efficient reasoning with hidden thinking. *arXiv preprint:2501.19201*, 2025. 2

[54] Gemma Team, Thomas Mesnard, Cassidy Hardin, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint:2407.21783*, 2024. 2

[55] Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. Llama: Open and efficient foundation language models. *arXiv preprint:2302.13971*, 2023. 1

[56] Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint:2307.09288*, 2023. 2

[57] Peifang Wang, Olga Golovneva, Armen Aghajanyan, et al. DOMINO: A dual-system for multi-step visual language reasoning. *arXiv preprint:2310.02804*, 2023. 2

[58] Peng Wang, Shuai Bai, Sinan Tan, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint:2409.12191*, 2024. 1, 2, 5, 6, 7

[59] Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *proceeding of NeurIPS*, 37:113569–113697, 2025. 1

[60] Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. Chain-of-thought prompting elicits reasoning in large language models. In *proceedings of NeurIPS*, pages 24824–24837, 2022. 1, 2

[61] Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint:2412.10302*, 2024. 2, 5, 6

[62] Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, et al. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *arXiv preprint:2402.12185*, 2024. 2, 5, 6

[63] Guowei Xu, Peng Jin, Li Hao, Yibing Song, et al. Llava-cot: Let vision language models reason step-by-step. *arXiv preprint:2411.10440*, 2024. 2

[64] Peixin Xu, Yujuan Ding, and Wenqi Fan. Chartadapter: Large vision-language model for chart summarization. *arXiv preprint:2412.20715*, 2024. 2

[65] Zhengzhuo Xu, Sinan Du, Yiyan Qi, Chengjin Xu, Chun Yuan, and Jian Guo. Chartbench: A benchmark for complex visual reasoning in charts. *arXiv preprint:2312.15915*, 2023. 1, 4, 6

[66] Zhengzhuo Xu, Bowen Qu, Yiyan Qi, Sinan Du, et al. Chartmoe: Mixture of expert connector for advanced chart under-

standing. *arXiv preprint:2409.03277*, 2024. 1, 2, 3, 4, 5, 6, 7

[67] Le Xue, Manli Shu, Anas Awadalla, Jun Wang, et al. xgen-mm (blip-3): A family of open large multimodal models. *arXiv preprint:2408.08872*, 2024. 2

[68] Pengyu Yan, Mahesh Bhosale, Jay Lal, Bikhyat Adhikari, and David S. Doermann. Chartreformer: Natural language-driven chart image editing. In *proceedings of ICDAR*, pages 453–469, 2024. 1, 2

[69] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, et al. Qwen2 technical report. *arXiv preprint:2407.10671*, 2024. 2, 5

[70] An Yang, Baosong Yang, Beichen Zhang, et al. Qwen2.5 technical report. *arXiv preprint:2412.15115*, 2024. 2, 3, 5

[71] Xudong Yang, Yifan Wu, Yizhang Zhu, Nan Tang, and Yuyu Luo. Askchart: Universal chart understanding through textual enhancement. *arXiv preprint:2412.19146*, 2024. 2

[72] Jiabo Ye, Anwen Hu, Haiyang Xu, et al. Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. In *Findings of ACL*, 2023. 2

[73] Qinghao Ye, Haiyang Xu, Guohai Xu, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint:2304.14178*, 2023. 2

[74] Qinghao Ye, Haiyang Xu, Jiabo Ye, et al. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *arXiv preprint:2311.04257*, 2023. 2, 5

[75] Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, et al. Tinychart: Efficient chart understanding with program-of-thoughts learning and visual token merging. In *proceedings of EMNLP*, pages 1882–1898, 2024. 1, 2, 5, 6

[76] Lingling Zhang, Muye Huang, QianYing Wang, Yaxian Wang, et al. Got-cqa: Graph-of-thought guided compositional reasoning for chart question answering. *arXiv preprint:2409.02611*, 2024. 2

[77] Pan Zhang, Xiaoyi Dong Bin Wang, Yuhang Cao, Chao Xu, et al. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint:2309.15112*, 2023. 2

[78] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint:2401.02385*, 2024. 5

[79] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, et al. Multimodal chain-of-thought reasoning in language models. *TMLR*, 2024, 2024. 1, 2

[80] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In *proceedings of NeurIPS*, 2023. 5

[81] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, et al. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *proceedings of ACL*, 2024. 5

[82] Tianyang Zhong, Zhengliang Liu, Yi Pan, Yutong Zhang, et al. Evaluation of openai o1: Opportunities and challenges of agi. *arXiv preprint:2409.18486*, 2024. 1, 2, 8

[83] Xun Zhu, Zheng Zhang, Xi Chen, Yiming Shi, Miao Li, and Ji Wu. Connector-s: A survey of connectors in multi-modal large language models. *arXiv preprint:2502.11453*, 2025. 2

[84] Li Zhuowan, Jasani Bhavan, Tang Peng, and Ghadar Shabnam. Synthesize step-by-step: Tools, templates and llms as data generators for reasoning-based chart vqa. In *proceedings of CVPR*, 2024. 2