Learning More from Less: Exploiting Counterfactuals for Data-Efficient Chart Understanding

Jianzhu Bao[1], Haozhen Zhang[1], Kuicai Dong[1], Bozhi Wu[1], Sarthak Ketanbhai Modi[1], Zi Pong Lim[2], Yon Shin Teo[2], Wenya Wang[1]*,

[1]Nanyang Technological University, [2]Aumovio Singapore Pte. Ltd.,
jianzhubao@gmail.com, wangwy@ntu.edu.sg

Abstract

Vision-Language Models (VLMs) have demonstrated remarkable progress in chart understanding, largely driven by supervised fine-tuning (SFT) on increasingly large synthetic datasets. However, scaling SFT data alone is inefficient and overlooks a key property of charts: charts are programmatically generated visual artifacts, where small, code-controlled visual changes can induce drastic shifts in semantics and correct answers. Learning this counterfactual sensitivity requires VLMs to discriminate fine-grained visual differences, yet standard SFT treats training instances independently and provides limited supervision to enforce this behavior. To address this, we introduce ChartCF, a data-efficient training framework designed to enhance counterfactual sensitivity. ChartCF consists of: (1) a counterfactual data synthesis pipeline via code modification, (2) a chart similarity-based data selection strategy that filters overly difficult samples for improved training efficiency, and (3) multimodal preference optimization across both textual and visual modalities. Experiments on five benchmarks show that ChartCF achieves superior or comparable performance to strong chart-specific VLMs while using significantly less training data.[1]

1 Introduction

Chart understanding is a critical capability for VLMs, serving as a cornerstone for automated data analysis, document understanding, and scientific research (Masry et al., 2025b; He et al., 2024; Xu et al., 2024b; Dong et al., 2025c,b; Masry et al., 2025d). Given a chart image, VLMs must accurately extract important values, identify underlying trends, and perform complex reasoning to answer user questions. To rigorously evaluate this capability, the research community has developed comprehensive benchmarks such as ChartQA (Masry et al., 2022), ChartX (Xia et al., 2025), and CharXiv (Wang et al., 2024b). Despite the rapid progress of proprietary models like GPT-4o, open-source chart-specific VLMs still exhibit significant performance gaps on these challenging benchmarks. They often struggle to locate subtle yet critical details and values in charts, leading to inferior extraction and reasoning performance compared to their proprietary counterparts. Closing this gap is particularly important for real-world applications that involve sensitive data (e.g., financial reports, medical records) or large-scale chart processing, where locally deployable open-source VLMs are often preferred over proprietary APIs for privacy and cost considerations.

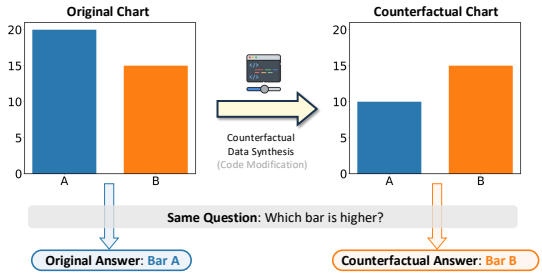


Figure 1: Illustration of a counterfactual pair. Given the same question "*Which bar is higher?*", a small modification to the height of bar A (left: original chart, right: counterfactual chart) results in a different answer.

The prevailing approach to addressing this gap has focused on scaling up training data through sophisticated synthesis pipelines. Recent work (Yang et al., 2025b; He et al., 2025) leverages code-based rendering tools (e.g., Matplotlib) and advanced LLMs to synthesize large-scale data for chart question answering, covering diverse topics and visual styles. While supervised fine-tuning (SFT) on these datasets effectively improves VLMs' chart understanding capability, it is data-inefficient and funda-

* Corresponding Author
[1]Code is available at https://github.com/jianzhubao/ChartCF.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 27775–27797
July 2-7, 2026

mentally overlooks a unique property of charts that demands more targeted supervision. Unlike natural images in other multimodal tasks (Wang et al., 2024a; Xiao et al., 2025), charts are programmatically generated visual artifacts where code controls all visual elements: from data values and trends to colors, labels, styles, etc. A small modification to any answer-critical element can alter the visual display and shift the semantic interpretation. This ultimately leads to a different correct answer. As illustrated in Figure 1, reducing the height of bar A causes the answer to "Which bar is higher?" to change from A to B, despite the overall visual similarity between the two charts. This *counterfactual sensitivity* requires VLMs to discriminate fine-grained visual differences. However, standard SFT treats each chart-question pair independently, providing no explicit supervision for such discriminative behavior. Consequently, even VLMs trained on massive SFT data may hallucinate when subtle visual differences are critical.

To address this limitation, we introduce counterfactual supervision for chart understanding. Here, a counterfactual refers to a chart instance obtained through a minimal, code-controlled modification that preserves overall appearance while inducing a different correct answer. Building on this idea, we propose ChartCF, a data-efficient training framework designed to leverage such counterfactual chart pairs through preference optimization. It consists of three key components: (i) a counterfactual data synthesis pipeline, (ii) a chart similarity-based data selection strategy, and (iii) a contrastive preference optimization method.

First, to construct counterfactual data, we leverage existing high-quality synthetic datasets (Yang et al., 2025b) and employ an advanced VLM to programmatically modify only answer-critical elements in the underlying plotting code. This ensures that counterfactual charts remain visually similar while inducing different ground-truth answers. Second, to improve training efficiency, we introduce a chart similarity-based data selection strategy that filters overly difficult samples, as they can introduce noise during preference learning. Third, we apply Direct Preference Optimization (Rafailov et al., 2023) across both modalities: Text DPO trains the model to favor answers corresponding to the presented chart while rejecting answers of counterfactual charts, whereas Image DPO associates each answer with its correct chart. This joint optimization effectively grounds answers in precise visual evidence.

We validate ChartCF on 5 widely-adopted benchmarks and results show that ChartCF achieves superior or comparable performance to strong open-source chart-specific VLMs. Notably, compared to the ECD baseline (Yang et al., 2025b) trained on 300K samples, our approach achieves comparable performance with only 4K preference pairs.

- We introduce a data synthesis pipeline to generate *counterfactual chart* pairs, coupled with a similarity-based selection strategy for improved training efficiency.
- We explore multimodal preference learning as a data-efficient alternative to SFT for chart understanding, demonstrating its effectiveness across multiple optimization objectives.
- Extensive experiments demonstrate that ChartCF achieves strong results while using significantly less training data, establishing a new paradigm for data-efficient chart understanding.

2 Related Work

Chart understanding (Huang et al., 2024) encompasses several prominent tasks, such as chart question answering (Kafle et al., 2018; Methani et al., 2020a; Masry et al., 2022, 2025a), chart-to-code generation (Liu et al., 2023; Zhao et al., 2025; Yang et al., 2025a), captioning (Obeid and Hoque, 2020; Kantharaj et al., 2022b), and retrieval (Dong et al., 2025a). Among these, chart question answering has emerged as a focal point of research due to its comprehensive nature, requiring models to integrate both visual perception and complex reasoning (Kafle et al., 2018; Wu et al., 2023; Wang et al., 2024b; He et al., 2025). Consequently, we primarily focus on chart question answering in this paper.

In recent years, a surge of benchmarks has been introduced to evaluate the chart question answering task (Methani et al., 2020a; Kantharaj et al., 2022a; Xia et al., 2025; Xu et al., 2024a). These datasets generally fall into two categories: real-world and synthetic. Real-world benchmarks, such as ChartQA (Masry et al., 2022) and CharXiv (Wang et al., 2024b), provide authentic and diverse visualizations sourced from business reports and scientific publications, capturing the complexity of human-designed charts. Conversely, synthetic benchmarks like PlotQA (Methani et al., 2020b) and ReachQA (He et al., 2025) leverage programmatic tools and LLMs to generate diverse charts. These benchmarks have evolved from descriptive

tasks to sophisticated multi-step reasoning over visual elements.

From a methodological perspective, a major research trend focuses on developing specialized chart models through large-scale training data synthesis. By leveraging code-based rendering tools and advanced LLMs, researchers have curated massive training datasets that emphasize both scale and diversity (Xu et al., 2024b; Fan et al., 2025; Masry et al., 2025c; Tang et al., 2025a; Huang et al., 2025a,b). By applying SFT on these datasets, various chart-specific VLMs have been developed, exhibiting strong domain-specific performance (Han et al., 2023; Zhang et al., 2024; Masry et al., 2024; Jiang and Luo, 2025; Masry et al., 2025d; He et al., 2025; Yang et al., 2025b). Some recent work has explored reinforcement learning to further enhance reasoning capabilities (Huang et al., 2025c; Sinha et al., 2025). However, these approaches typically still require extensive SFT training on large-scale datasets before RL (Chen et al., 2025; Liu et al., 2025). In parallel, another line of research explores training-free approaches that leverage prompt engineering and external tools to enhance chart understanding (Wang et al., 2025a; Kaur et al., 2025).

Unlike these approaches that focus on scaling up training data, ChartCF emphasizes supervision quality through counterfactual data. By providing explicit contrastive supervision on visually similar charts with different answers, combined with similarity-based data selection, ChartCF achieves effective training with significantly less data.

3 Method

We present ChartCF, a framework designed to boost the performance of VLMs in chart understanding through counterfactual-oriented training. Figure 2 shows the architecture of ChartCF. It first constructs a visual contrastive dataset through an automated counterfactual data synthesis pipeline, followed by a simple and efficient data selection strategy. Finally, ChartCF performs preference optimization targeting both textual responses and the chart images, improving the VLMs' capability to correctly associate visually similar charts with their respective answers.

3.1 Problem Definition

The goal of chart question answering is to generate a textual response y given a chart image I and a natural language question x. Formally, a VLM models the conditional probability $P_\theta(y|I, x)$, where θ represents the model parameters.

3.2 Counterfactual Data Synthesis

To construct training data for preference optimization, we propose a counterfactual data synthesis pipeline. Instead of generating charts from scratch, we leverage the existing high-quality synthetic dataset, ECD (Yang et al., 2025b), to create visually similar contrastive pairs with discriminative yet small differences through code-level modification.

Initial Data. We start with a dataset of seed samples, where each sample is a quadruplet $S_o = (I_o, C_o, Q, A_o)$. Here, I_o is the chart image, C_o is the executable plotting code (e.g., Python Matplotlib script) used to render I_o, Q is a question related to the chart, and A_o is the corresponding ground-truth answer.

Counterfactual Code Generation. For each seed sample, we employ an advanced VLM (e.g., GPT-5) as a code modifier. The goal is to generate a modified code snippet C_c that results in a different answer for the same question Q, while maintaining overall visual similarity to the original chart. Our prompt[2] design requires the advanced VLM to first isolate the specific code elements directly responsible for the current answer A_o—such as a specific data value, a subplot title, or a category label. The VLM is then instructed to modify only these answer-critical elements to produce A_c, while keeping all other components, including unrelated data points, colors, and random seeds, strictly identical to the original code C_o. This targeted modification process ensures that the difference between the original and counterfactual charts is isolated to the information required to answer the question, creating a precise signal for preference optimization.

Rendering and Pairing. We execute the modified code C_c to render the counterfactual chart image I_c, yielding a new sample $S_c = (I_c, C_c, Q, A_c)$. Crucially, since the code modifications are strictly confined to answer-critical elements, I_c maintains overall visual similarity to the original image I_o. By pairing the original sample S_o with this counterfactual sample S_c, we construct a counterfactual pair:

$$\mathcal{D}_{\text{pair}} = \{(I_o, A_o), (I_c, A_c)|Q\} \quad (1)$$

[2]Prompts are shown in Figures 4 and 5 of Appendix A.

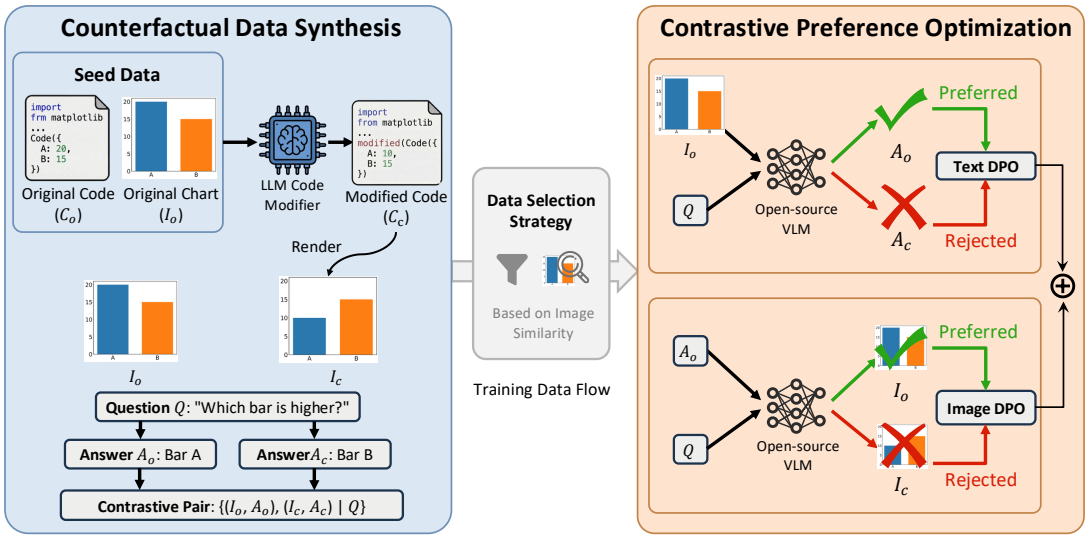


Figure 2: Overview of ChartCF. **Left**: Counterfactual data synthesis pipeline that generates counterfactual chart pairs (I_o, I_c) with their respective answers A_o and A_c via code-level modification. **Middle**: Chart similarity-based data selection that filters overly difficult pairs. **Right**: Contrastive preference optimization using Text DPO and Image DPO to train the model to distinguish between visually similar charts with different answers.

where A_c is the ground-truth answer for I_c. This synthesized data provides the necessary positive and negative signals for the subsequent preference optimization. The quality of the synthesized data is further validated through a multi-stage pipeline (Appendix B).

3.3 Data Selection Strategy

Recent research suggests that preference learning algorithms, such as DPO, are sensitive to the difficulty level of training samples; overly difficult samples can prove detrimental rather than beneficial, potentially degrading optimization performance (Gou and Nguyen, 2025; Gao et al., 2025). Inspired by this, we propose a simple yet effective data selection strategy based on chart image similarity. Specifically, following previous work (Tang et al., 2025b; Yang et al., 2025a), we employ GPT-5-mini to quantify the visual similarity score between the original chart I_o and the counterfactual chart I_c.[3] Intuitively, a higher similarity score implies that the visual modification is more subtle and thus harder to perceive, creating a hard negative sample. Consequently, we rank the generated pairs by their image similarity scores and select the $\rho\%$ of samples with the lowest similarity scores (i.e., filtering out those samples with overly subtle visual changes). This strategy eliminates "noisy hard" samples, allowing us to achieve superior data efficiency and model performance with a smaller, higher-quality dataset.

3.4 Contrastive Preference Optimization

We employ Direct Preference Optimization (DPO) (Rafailov et al., 2023) to align the model with the constructed counterfactual preference data. Standard DPO typically focuses solely on optimizing preferences over textual responses. However, we argue that for chart understanding, visual discrimination is equally critical. Thus, following recent work on multimodal preference alignment (Wang et al., 2024a; Wu et al., 2025), we adopt a dual-alignment strategy that optimizes model preferences across both textual and visual modalities, formulated as a joint optimization of **Text DPO** and **Image DPO**.

Text DPO. Text DPO optimizes preferences at the textual response level. Given the original chart image I_o and question Q, the model should favor the ground-truth answer A_o while rejecting the counterfactual answer A_c. Note that A_c is a particularly challenging negative sample because it is valid for the visually similar chart I_c but factually incorrect for I_o. We denote the policy model as π_θ and the reference model as π_{ref}. Following the standard DPO formulation (Rafailov et al., 2023), the Text DPO loss is defined as:

$$\mathcal{L}_{\text{text-dpo}}(I_o, Q, A_o, A_c) = -\log\sigma\left(\beta\log\frac{\pi_\theta(A_o|I_o,Q)}{\pi_{\text{ref}}(A_o|I_o,Q)} - \beta\log\frac{\pi_\theta(A_c|I_o,Q)}{\pi_{\text{ref}}(A_c|I_o,Q)}\right) \quad (2)$$

[3]The prompt is shown in Figure 6 of Appendix A.

where β is the temperature parameter controlling the strength of the KL constraint. By minimizing this loss, we penalize the model for generating answers that correspond to counterfactual visual states (A_c) when presented with the actual image (I_o), thereby sharpening its ability to distinguish fine-grained semantic differences.

Image DPO. Image DPO optimizes preferences at the visual input level, providing a complementary perspective to Text DPO. It trains the VLM to identify the correct visual context for a given answer. Specifically, we fix the question Q and the original ground-truth answer A_o, and require the model to assign a higher likelihood to A_o when conditioned on the correct image I_o compared to the counterfactual image I_c. The Image DPO loss is defined as:

$$\mathcal{L}_{\text{img-dpo}}(I_o, I_c, Q, A_o) = \\ -\log\sigma\left(\beta\log\frac{\pi_\theta(A_o|I_o,Q)}{\pi_{\text{ref}}(A_o|I_o,Q)} - \beta\log\frac{\pi_\theta(A_o|I_c,Q)}{\pi_{\text{ref}}(A_o|I_c,Q)}\right) \tag{3}$$

Overall Training Objective. The final training objective combines the two preference losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{text-dpo}} + \mathcal{L}_{\text{img-dpo}} \tag{4}$$

By jointly optimizing these two objectives, ChartCF trains the model to not only select the correct answer for a given chart but also associate each answer with its corresponding visual evidence. We note that ChartCF's counterfactual data is broadly compatible with other multimodal preference optimization objectives, such as mDPO (Wang et al., 2024a) and S-VCO (Wu et al., 2025), as shown in Section 4.8.

4 Experiments

4.1 Experimental Setup

Training Data. We utilize the training data of ECD (Yang et al., 2025b) as our data source, which provides over 10K chart images with Python plotting code and 300K QA pairs. To construct our preference training set, we use all 10K chart images and randomly select one QA pair per image as seed samples. We then employ gpt-5-2025-08-07 to modify the plotting code and generate a corresponding new answer. This pipeline yields a training set of 10K preference pairs. In the main experiments, we apply our data selection strategy to retain 4K pairs for training. The impact of different data retention ratios is discussed in Section 4.5.

Evaluation Benchmarks. We conduct experiments across five widely-adopted benchmarks, encompassing both real-world charts (CharXiv (Wang et al., 2024b), ChartQA (Masry et al., 2022)) and synthetic charts (ChartBench (Xu et al., 2024a), ChartX (Xia et al., 2025), ECDBench (Yang et al., 2025b)). Detailed descriptions of each benchmark are provided in Appendix C. We primarily use CharXiv for detailed analysis experiments, due to its realism, complexity, and broad adoption.

Baselines. We primarily compare against chart-specific VLMs, including ChartGemma (Masry et al., 2025d), TinyChart (Zhang et al., 2024), ChartReasoner (Jia et al., 2025), Chart-R1 (Chen et al., 2025), ECD (Yang et al., 2025b), and ReachQA (He et al., 2025). Additionally, we evaluate general-purpose open-source VLMs as base models, including InternVL3.5-8B-Instruct (Wang et al., 2025b), Qwen2.5-VL-7B-Instruct (Bai et al., 2025), Qwen3-VL-8B-Instruct (Team, 2025), reporting their performance both before and after training with ChartCF. Furthermore, we provide the performance of several proprietary models as reference points, including GPT-4o mini, GPT-4o (OpenAI et al., 2024), Claude-3.5-Sonnet (Anthropic, 2024), GPT-5 (OpenAI, 2025), and Gemini-2.5-Pro (Comanici et al., 2025).

Evaluation Protocols. We follow ECD (Yang et al., 2025b)[4] to ensure a fair comparison. GPT-4o is employed to evaluate answer correctness against ground-truth references for all benchmarks except ChartBench "yes/no" questions, where we use regular expression matching.

Implementation Details. We employ Low-Rank Adaptation (LoRA) (Hu et al., 2022) for efficient DPO fine-tuning. Specifically, InternVL3.5-8B-Instruct, Qwen2.5-VL-7B-Instruct, and Qwen3-VL-8B-Instruct are fine-tuned with a LoRA rank of 64, alpha of 64, a batch size of 64, and a learning rate of 1e-4 for 2 epochs on 8 A100 80GB GPUs. Following common practice in VLM fine-tuning, we freeze the vision encoder and projection layers, updating only the language model parameters during training.

4.2 Main Results

We present the main results on real-world and synthetic chart benchmarks in Tables 1 and 2, respec-

[4] https://github.com/yuweiyang-anu/ECD/tree/main/evaluation

Method	Model Size	# Training Data	CharXiv			ChartQA
			Descriptive	Reasoning	Average	
Proprietary VLMs						
GPT-4o mini†	-	-	74.92	34.10	66.76	77.52
GPT-4o†	-	-	84.45	47.10	76.98	85.70
Claude-3.5-Sonnet†	-	-	84.30	60.20	79.48	90.80
GPT-5	-	-	90.03	69.10	85.84	87.60
Gemini-2.5-Pro	-	-	91.22	68.40	86.66	89.68
Chart-specific/Open-Source VLMs						
TinyChart†	3B	1.36M (SFT)	-	8.30	-	83.60
ChartGemma†	3B	123K (SFT)	21.30	12.50	19.54	80.16
ChartReasoner†	7B	140K (SFT)	-	-	-	86.93
Chart-R1†	7B	228K (SFT) + 30K (RL)	62.00	<u>46.20</u>	58.84	**91.04**
ReachQA (InternVL2-8B)†	8B	20K (SFT)	54.83	32.70	50.40	82.44
InternVL3.5	8B	-	75.00	40.00	68.00	86.88
+ ECD	8B	300K (SFT)	77.03	40.30	69.68	86.52
+ ChartCF (Ours)	8B	**4K** pairs (DPO)	80.47	42.30	72.84	<u>87.16</u>
Qwen2.5-VL†	7B	-	66.40	41.20	61.36	83.04
+ ECD†	7B	300K (SFT)	74.20	40.20	67.40	85.32
+ ChartCF (Ours)	7B	**4K** pairs (DPO)	75.08	44.40	68.94	87.00
Qwen3-VL	8B	-	81.25	43.10	73.62	85.44
+ ECD	8B	300K (SFT)	<u>81.93</u>	42.20	<u>73.98</u>	85.12
+ ChartCF (Ours)	8B	**4K** pairs (DPO)	**82.58**	**46.50**	**75.36**	85.24

Table 1: Performance comparison on real-world chart benchmarks: CharXiv and ChartQA. Results marked with † are taken from prior work (Yang et al., 2025b; He et al., 2025; Chen et al., 2025); other results are our own implementations. For chart-specific models, we report the number of training samples (# Training Data), where SFT, RL, and DPO denote the amount of data used for SFT, reinforcement learning, and DPO, respectively. The best results among chart-specific and open-source VLMs are in **bold**, and the second-best are <u>underlined</u>.

tively. Overall, ChartCF consistently improves the performance of various base models (InternVL3.5, Qwen2.5-VL, and Qwen3-VL), achieving comparable or superior performance to strong open-source baselines across five benchmarks while using significantly less training data. These results highlight that, compared to data-intensive SFT baselines, ChartCF achieves the most favorable trade-off between data efficiency and performance.

Specifically, among open-source models, ChartCF achieves the best average performance on CharXiv when applied to Qwen3-VL, particularly excelling on the most challenging reasoning questions (Table 1). It also outperforms ECD on CharXiv across all three base models (Qwen2.5-VL, Qwen3-VL, InternVL3.5). While ChartCF achieves lower scores than Chart-R1 on ChartQA, Chart-R1 requires extensive training with 228K SFT samples plus 30K RL samples, with training times of 3 hours for SFT and 30 hours for RL on 24 H800 GPUs (Chen et al., 2025). In contrast, our method requires only 4K preference pairs for DPO training, which takes approximately 40 minutes on 8 A100 GPUs, resulting in substantially lower computational costs and training time. According to Table 2, on ChartBench, ChartX, and ECDBench, ChartCF achieves competitive or superior performance compared to ECD across various base models while using substantially less training data, further demonstrating data efficiency.

Beyond the standard "Text DPO + Image DPO" formulation, we further show in Section 4.8 that ChartCF is compatible with alternative multimodal preference optimization objectives, with all variants achieving comparable performance.

4.3 Enhancing SFT Baselines

To investigate whether ChartCF can further enhance VLMs already trained with large-scale SFT data, we apply our method on top of ECD-trained VLMs. Results in Table 3 show that ChartCF consistently improves performance across both Qwen2.5-VL and Qwen3-VL. These results demonstrate that ChartCF provides complementary benefits to SFT, suggesting that counterfactual supervision and SFT capture different aspects of chart understanding. This compatibility highlights ChartCF's practical value as a versatile enhancement for existing chart-specific VLMs.

4.4 Ablation Studies

We conduct ablation studies on CharXiv and ChartQA using Qwen2.5-VL as the base model. Results are shown in Table 4. First, removing Text DPO, Image DPO, or the data selection strategy individually leads to performance degradation

Method	Model Size	# Training Data	ChartBench			ChartX	ECDBench		
			Binary	NQA	Avg.		Des.	Rea.	Avg.
Proprietary VLMs									
GPT-4o mini†	-	-	70.26	34.93	74.33	44.36	57.27	24.26	40.77
GPT-4o†	-	-	81.03	52.88	77.90	58.33	70.18	35.62	52.90
Claude-3.5-Sonnet†	-	-	76.72	48.29	73.56	42.71	68.14	41.99	55.07
GPT-5	-	-	-	79.33	-	83.51	78.27	63.15	70.71
Gemini-2.5-Pro†	-	-	-	71.24	-	74.22	76.88	44.36	60.62
Chart-specific/Open-Source VLMs									
ChartGemma†	3B	123K (SFT)	78.90	34.10	73.92	35.15	-	-	-
ChartReasoner†	7B	140K (SFT)	-	-	55.20	-	-	-	-
ReachQA (InternVL2-8B)†	8B	20K (SFT)	65.90	47.29	63.83	45.38	-	-	-
InternVL3.5	8B	-	79.42	67.71	78.12	68.49	49.67	32.27	40.97
+ ECD	8B	300K (SFT)	79.00	67.33	77.70	70.31	65.77	39.79	52.78
+ ChartCF (Ours)	8B	**4K** pairs (DPO)	79.87	67.95	78.55	68.40	65.28	38.89	52.08
Qwen2.5-VL†	7B	-	80.99	67.81	79.53	67.80	57.35	19.04	38.19
+ ECD†	7B	300K (SFT)	79.35	70.86	78.41	70.83	66.34	35.38	50.86
+ ChartCF (Ours)	7B	**4K** pairs (DPO)	82.45	**72.43**	81.34	**73.18**	66.09	37.42	51.76
Qwen3-VL	8B	-	71.43	72.33	71.53	67.80	**71.65**	41.83	56.74
+ ECD	8B	300K (SFT)	81.42	71.95	80.37	71.61	70.51	**43.22**	**56.87**
+ ChartCF (Ours)	8B	**4K** pairs (DPO)	**83.36**	71.95	**82.09**	72.14	70.67	42.08	56.38

Table 2: Performance comparison on synthetic chart benchmarks: ChartBench, ChartX, and ECDBench. For ChartBench, "Binary" denotes yes/no questions and "NQA" denotes numerical questions. For ECDBench, "Des." and "Rea." refer to descriptive and reasoning questions, respectively. "Avg." is short for "Average".

Method	CharXiv			ChartQA
	Des.	Rea.	Avg.	
Qwen2.5-VL				
ChartCF Only (4K DPO)	75.08	44.40	68.94	**87.00**
ECD (300K SFT)	74.20	40.20	67.40	85.32
+ ChartCF (300K SFT + 4K DPO)	**81.20**	**46.10**	**74.18**	85.48
Qwen3-VL				
ChartCF Only (4K DPO)	82.58	46.50	75.36	85.24
ECD (300K SFT)	81.93	42.20	73.98	85.12
+ ChartCF (300K SFT + 4K DPO)	**83.45**	**48.40**	**76.44**	**86.76**

Table 3: Performance gains from applying ChartCF on top of ECD-trained models.

Method	CharXiv			ChartQA
	Des.	Rea.	Avg.	
ChartCF (Ours)	**75.08**	**44.40**	**68.94**	**87.00**
w/o Text DPO	73.50	42.80	67.36	83.88
w/o Image DPO	72.67	43.00	66.74	84.56
w/o Data Selection	73.08	43.20	67.10	84.88
SFT w. 4K Samples	72.55	39.20	65.88	84.76
SFT w. 8K Samples	71.50	39.40	65.08	85.20

Table 4: Ablation study on CharXiv and ChartQA based on Qwen2.5-VL. We evaluate the contribution of each component in ChartCF and compare against SFT baselines trained with similar amounts of data.

across both benchmarks, demonstrating that dual optimization across both modalities combined with careful data selection is essential for ChartCF's success. Second, we compare ChartCF against SFT baselines trained on equivalent data: "SFT w. 4K Samples" uses the original ECD samples corresponding to our 4K counterfactual pairs, while "SFT w. 8K Samples" additionally includes the 4K counterfactual samples, matching ChartCF's total training instances (4K pairs = 8K individual samples). ChartCF consistently outperforms both SFT baselines, with particularly substantial improvements on reasoning questions, highlighting that explicit contrastive supervision is more effective than standard SFT.

4.5 Analysis of Data Selection

To validate our chart similarity-based data selection strategy, we conduct experiments on CharXiv with varying data retention ratios ρ, as shown in Figure 3. We compare our strategy ("Keep Low") against two baselines: random sampling and an inverse strategy that retains pairs with the highest similarity scores ("Keep High").

Our "Keep Low" strategy consistently outperforms random sampling and "Keep High" across various retention ratios, demonstrating that selecting pairs with lower similarity is essential for identifying valuable training samples. Notably, our method achieves optimal performance when $\rho = 40$, where only $\sim$4K pairs of training samples are included. The performance of "Keep Low" first increases as more data is included, then decreases at higher retention ratios. This reflects two competing effects: initially, too few samples lead to insufficient training, while at higher retention ratios, the inclusion of excessively difficult samples introduces noise that degrades performance.

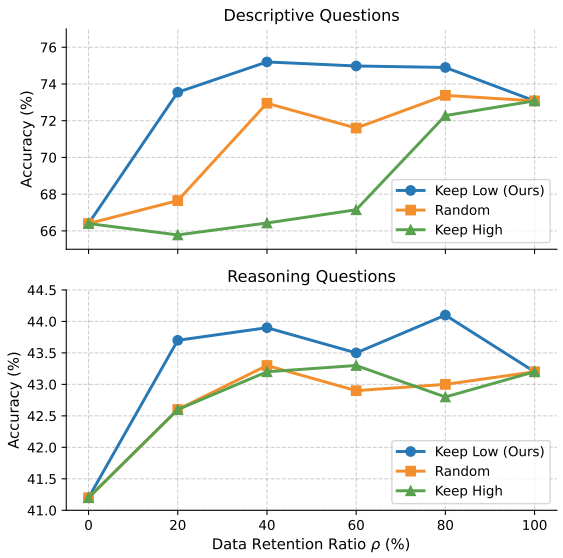


Figure 3: Impact of data selection strategy on CharXiv. We compare three selection methods across different data retention ratios ρ (x-axis): "Keep Low (Ours)" ranks counterfactual pairs by chart similarity and retains the $\rho\%$ pairs with the lowest similarity scores, "Random" randomly retains pairs, and "Keep High" retains pairs with the highest similarity scores.

In contrast, random sampling shows an increasing trend but fails to reach the peak performance of our method due to persistent noisy samples at all ratios. Finally, "Keep High" performs poorly, especially on descriptive questions, as excessively difficult samples act as noise. These results validate our hypothesis that preference learning is sensitive to sample difficulty. Furthermore, filtering based on chart similarity provides an effective mechanism to balance training data quality and quantity.

4.6 Analysis of Different Types of Hard Samples

To better understand what makes a counterfactual pair difficult for preference optimization, we investigate two types of synthetic data. The first type, "Synthetic Data w/o Distractors", follows our standard pipeline in Section 3.2, modifying only answer-critical elements. The second type, "Synthetic Data w. Distractors", additionally modifies non-critical visual elements (e.g., unrelated data points, colors) that do not affect the answer but increase overall visual differences. For each type, we examine different chart similarity-based selections: the lowest half and the highest half.

Results in Table 5 reveal two categories of hard samples that hinder DPO training. First, within the clean synthetic data without distractors, high-

Method	CharXiv		
	Des.	Rea.	Avg.
Qwen2.5-VL	66.40	41.20	61.36
Synthetic Data w/o Distractors			
ChartCF w/o Data Selection (10K)	73.08	43.20	67.10
w. Lowest Half Pairs (5K)	**75.65**	**43.70**	**69.26**
w. Highest Half Pairs (5K)	66.42	43.30	61.80
Synthetic Data w. Distractors			
ChartCF w/o Data Selection (10K)	69.92	43.10	64.56
w. Lowest Half Pairs (5K)	64.60	38.90	56.26
w. Highest Half Pairs (5K)	66.80	40.00	61.44

Table 5: Analysis of different types of hard samples on CharXiv. We compare clean synthetic data (modifications only to answer-critical elements) versus data with distractors (additional modifications to non-critical elements), across different similarity-based selections.

similarity pairs (Highest Half) prove detrimental, significantly degrading performance compared to low-similarity pairs (Lowest Half). This confirms our hypothesis in Section 3.3 that overly subtle modifications create noisy signals for training. Second, adding distractors uniformly hurts performance across all similarity levels. This suggests that distractors introduce spurious correlations that mislead the preference optimization process, as the model struggles to identify which visual differences are answer-critical versus merely distracting. These findings highlight that effective preference learning requires counterfactual pairs that isolate answer-critical differences while remaining visually distinguishable, as both overly subtle changes and irrelevant modifications impede learning.

4.7 Compatibility with Different Code Modifiers

To verify the compatibility of ChartCF with different code modifiers, we conduct a small-scale experiment using 2K counterfactual pairs synthesized by various code modifiers, including four proprietary VLMs (GPT-5, GPT-4o, Gemini-2.5-Pro, and Claude-4.5-Sonnet) and three open-source alternatives from the Qwen3-VL-Thinking family (8B, 32B, and 235B-A22B). As shown in Table 6, ChartCF yields consistently strong performance across all modifiers. Notably, even compact open-source models such as Qwen3-VL-8B-Thinking and Qwen3-VL-32B-Thinking achieve substantial gains over the base Qwen2.5-VL, and the larger Qwen3-VL-235B-A22B-Thinking closes the gap further, reaching performance competitive with proprietary alternatives such as GPT-5. These findings suggest that the effectiveness of ChartCF is primarily driven by the counterfactual data construction

Method	CharXiv		
	Des.	Rea.	Avg.
Qwen2.5-VL	66.40	41.20	61.36
Proprietary Code Modifiers			
+ ChartCF w. GPT-5	73.55	43.70	67.58
+ ChartCF w. GPT-4o	73.43	42.20	67.18
+ ChartCF w. Gemini-2.5-Pro	75.00	43.30	68.66
+ ChartCF w. Claude-4.5-Sonnet	**75.88**	**44.00**	**69.50**
Open-Source Code Modifiers			
+ ChartCF w. Qwen3-VL-235B-A22B-Thinking	74.35	42.90	68.06
+ ChartCF w. Qwen3-VL-32B-Thinking	74.30	42.20	67.88
+ ChartCF w. Qwen3-VL-8B-Thinking	72.35	41.80	66.24

Table 6: Performance comparison on CharXiv using different proprietary and open-source VLMs as code modifiers, trained on a 2K sample subset.

Method	CharXiv			ChartQA
	Des.	Rea.	Avg.	
ECD (300K SFT)	74.20	40.20	67.40	85.32
ChartCF (Ours)	75.08	**44.40**	68.94	87.00
mDPO (ChartCF + AncPO)	74.38	43.00	68.10	**87.28**
S-VCO	**75.85**	43.30	**69.34**	86.44
VCO (S-VCO w/o sym.)	75.25	41.10	68.42	86.04
ChartCF + S-VCO	74.55	44.00	68.44	87.08

Table 7: Compatibility of ChartCF's counterfactual data with alternative multimodal preference optimization objectives on CharXiv and ChartQA, all trained on Qwen2.5-VL with the same 4K counterfactual pairs. "VCO" denotes S-VCO without the symmetric term. "ChartCF + S-VCO" combines the two losses.

paradigm itself, rather than the capability of any specific proprietary model. Practitioners can adopt our pipeline with locally deployed open-source models, maintaining data privacy while achieving comparable performance at significantly lower cost.

4.8 Compatibility with Alternative Multimodal Preference Optimization Objectives

Beyond the standard "Text DPO + Image DPO" formulation, ChartCF is broadly compatible with other multimodal preference optimization objectives designed for fine-grained visual grounding. To demonstrate this, we apply two representative methods, mDPO (Wang et al., 2024a) and S-VCO (Wu et al., 2025), to our counterfactual data. Specifically, mDPO augments the "Text DPO + Image DPO" framework with an Anchored Preference Optimization (AncPO) loss, while S-VCO further introduces a symmetric visual contrastive term. For fairness, all methods are trained on Qwen2.5-VL using the same 4K counterfactual pairs as in our main experiments.

As shown in Table 7, ChartCF's counterfactual data works effectively with various preference optimization objectives, with all variants achieving comparable performance and consistently outperforming the 300K-sample SFT baseline.

4.9 Case Study

We present two types of case studies in the appendices. First, we showcase four counterfactual chart pairs generated by our synthesis pipeline, covering descriptive questions (legend interpretation, axis scale recognition) and reasoning questions (numerical computation, spatial relationships), with modifications including textual labels, numerical values, and visual element positions (Appendix E). Second, we compare ChartCF and ECD on five examples, demonstrating that ChartCF achieves superior performance on tasks requiring fine-grained element localization, precise value reading, spatial reasoning, and numerical computation (Appendix F).

4.10 Additional Analyses

In the appendix, we provide additional analyses on the cost and validation of our data synthesis pipeline (Appendix B), as well as a study on curriculum learning for hard samples (Appendix D).

5 Conclusion

We introduce ChartCF, a data-efficient training framework that enhances chart understanding through counterfactual learning. Unlike conventional approaches that rely on scaling training data, our method leverages the programmatic nature of charts to synthesize targeted counterfactual pairs via code modifications. Combined with similarity-based data selection and contrastive preference optimization across both textual and visual modalities, ChartCF achieves competitive or superior performance compared to strong chart-specific VLMs while using significantly less training data.

Limitations

While ChartCF demonstrates strong data efficiency, our work has several limitations that warrant future investigation.

First, our approach relies on the availability of executable plotting code. While synthetic chart datasets with code are increasingly available, extending our method to naturally occurring charts would require developing code generation or reverse-engineering techniques.

Second, our counterfactual synthesis pipeline depends on advanced VLMs (e.g., GPT-5) to gen-

erate modified code, which introduces additional costs during data preparation. However, we show in Section 4.7 that open-source models can serve as effective alternatives, significantly reducing the overhead of our method.

Ethics Statement

This work uses publicly available datasets that have been widely adopted in prior chart understanding research. Our use of these datasets, and all other software and resources, strictly complies with their respective licenses and intended purposes. The datasets are understood to be free of personally identifiable information and offensive content. AI assistants are employed solely for grammar checking and text polishing during manuscript preparation.

We acknowledge that using LLMs for counterfactual data synthesis may introduce potential risks such as inaccuracies in generated code and answers. However, our experimental results demonstrate consistent performance improvements across multiple benchmarks, validating the effectiveness of our method despite potential noise in synthesized data. Moreover, synthetic data generation remains significantly more cost-effective and scalable than manual annotation.

Acknowledgments

This research was performed at the AUMOVIO-NTU Corporate Lab. This research/project is supported by A*STAR under the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative (Award: I2501E0045), as well as cash and in-kind contribution from the industry partner(s).

References

Anthropic. 2024. Introducing claude 3.5 sonnet.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Lei Chen, Xuanle Zhao, Zhixiong Zeng, Jing Huang, Yufeng Zhong, and Lin Ma. 2025. Chart-r1: Chain-of-thought supervision and reinforcement for advanced chart reasoner. *Preprint*, arXiv:2507.15509.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Kuicai Dong, Yujing Chang, Derrick Goh Xin Deik, Dexun Li, Ruiming Tang, and Yong Liu. 2025a. MMDocIR: Benchmarking multimodal retrieval for long documents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 30971–31005, Suzhou, China. Association for Computational Linguistics.

Kuicai Dong, Yujing Chang, Shijie Huang, Yasheng Wang, Ruiming Tang, and Yong Liu. 2025b. Benchmarking retrieval-augmented multimodal generation for document question answering. *Preprint*, arXiv:2505.16470.

Kuicai Dong, Shurui Huang, Fangda Ye, Wei Han, Zhi Zhang, Dexun Li, Wenjun Li, Qu Yang, Gang Wang, Yichao Wang, Chen Zhang, and Yong Liu. 2025c. Doc-researcher: A unified system for multimodal document parsing and deep research. *Preprint*, arXiv:2510.21603.

Wan-Cyuan Fan, Yen-Chun Chen, Mengchen Liu, Alexander Jacobson, Lu Yuan, and Leonid Sigal. 2025. In-depth and in-breadth: Pre-training multimodal language models customized for comprehensive chart understanding. *Preprint*, arXiv:2507.14298.

Chengqian Gao, Haonan Li, Liu Liu, Zeke Xie, Peilin Zhao, and Zhiqiang Xu. 2025. Principled data selection for alignment: The hidden risks of difficult examples. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Qi Gou and Cam-Tu Nguyen. 2025. Mixed preference optimization: Reinforcement learning with data selection and better reference model. *Preprint*, arXiv:2403.19443.

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*.

Wei He, Zhiheng Xi, Wanxu Zhao, Xiaoran Fan, Yiwen Ding, Zifei Shan, Tao Gui, Qi Zhang, and Xuanjing Huang. 2025. Distill visual chart reasoning ability

from LLMs to MLLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 3224–3250, Suzhou, China. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen andin Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Kung-Hsiang Huang, Hou Pong Chan, May Fung, Haoyi Qiu, Mingyang Zhou, Shafiq Joty, Shih-Fu Chang, and Heng Ji. 2024. From pixels to insights: A survey on automatic chart understanding in the era of large foundation models. *IEEE Transactions on Knowledge and Data Engineering*, 37(5):2550–2568.

Muye Huang, Han Lai, Xinyu Zhang, Wenjun Wu, Jie Ma, Lingling Zhang, and Jun Liu. 2025a. Evochart: A benchmark and a self-training approach towards real-world chart understanding. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 3680–3688. AAAI Press.

Muye Huang, Han Lai, Xinyu Zhang, Wenjun Wu, Jie Ma, Lingling Zhang, and Jun Liu. 2025b. Evochart: A benchmark and a self-training approach towards real-world chart understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3680–3688.

Muye Huang, Lingling Zhang, Jie Ma, Han Lai, Fangzhi Xu, Yifei Li, Wenjun Wu, Yaqiang Wu, and Jun Liu. 2025c. Chartsketcher: Reasoning with multimodal feedback and reflection for chart understanding. *Preprint*, arXiv:2505.19076.

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. *Preprint*, arXiv:2506.10116.

Gongyao Jiang and Qiong Luo. 2025. Chart-coca: Self-improving chart understanding of vision lms via code-driven synthesis and candidate-conditioned answering. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management, CIKM 2025, Seoul, Republic of Korea, November 10-14, 2025*, pages 1168–1178. ACM.

Kushal Kafle, Brian L. Price, Scott Cohen, and Christopher Kanan. 2018. DVQA: understanding data visualizations via question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 5648–5656. Computer Vision Foundation / IEEE Computer Society.

Shankar Kantharaj, Xuan Long Do, Rixie Tiffany Ko Leong, Jia Qing Tan, Enamul Hoque, and Shafiq R. Joty. 2022a. Opencqa: Open-ended question answering with charts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11817–11837. Association for Computational Linguistics.

Shankar Kantharaj, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq R. Joty. 2022b. Chart-to-text: A large-scale benchmark for chart summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4005–4023. Association for Computational Linguistics.

Rachneet Kaur, Nishan Srishankar, Zhen Zeng, Sumitra Ganesh, and Manuela Veloso. 2025. Chartagent: A multimodal agent for visually grounded reasoning in complex chart question answering. *Preprint*, arXiv:2510.04514.

Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Martin Eisenschlos. 2023. Matcha: Enhancing visual language pretraining with math reasoning and chart derendering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12756–12770. Association for Computational Linguistics.

Zhuoming Liu, Xiaofeng Gao, Feiyang Niu, Qiaozi Gao, Liu Liu, and Robinson Piramuthu. 2025. Start: Spatial and textual learning for chart understanding. *Preprint*, arXiv:2512.07186.

Ahmed Masry, Mohammed Saidul Islam, Mahir Ahmed, Aayush Bajaj, Firoz Kabir, Aaryaman Kartha, Md. Tahmid Rahman Laskar, Mizanur Rahman, Shadikur Rahman, Mehrad Shahmohammadi, Megh Thakkar, Md. Rizwan Parvez, Enamul Hoque, and Shafiq Joty. 2025a. Chartqapro: A more diverse and challenging benchmark for chart question answering. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 19123–19151. Association for Computational Linguistics.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq R. Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2263–2279. Association for Computational Linguistics.

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, Abhay Puri, Masoud Hashemi, Juan A. Rodriguez, Megh Thakkar, Khyati Mahajan, Vikas Yadav, Sathwik Tejaswi Madhusudhan, Alexandre Piché, Dzmitry Bahdanau, Christopher Pal, David Vazquez, Enamul Hoque, Perouz Taslakian, Sai Rajeswar, and Spandana Gella. 2025c. Bigcharts-r1: Enhanced chart reasoning with visual reinforcement finetuning. *Preprint*, arXiv:2508.09804.

Ahmed Masry, Mehrad Shahmohammadi, Md. Rizwan Parvez, Enamul Hoque, and Shafiq Joty. 2024. Chartinstruct: Instruction tuning for chart comprehension and reasoning. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 10387–10409. Association for Computational Linguistics.

Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. 2025d. Chartgemma: Visual instruction-tuning for chart reasoning in the wild. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 625–643.

Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. 2020a. Plotqa: Reasoning over scientific plots. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 1516–1525. IEEE.

Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. 2020b. Plotqa: Reasoning over scientific plots. In *Proceedings of the ieee/cvf winter conference on applications of computer vision*, pages 1527–1536.

Jason Obeid and Enamul Hoque. 2020. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. In *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020, Dublin, Ireland, December 15-18, 2020*, pages 138–147. Association for Computational Linguistics.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI. 2025. Gpt-5 is here.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Sanchit Sinha, Oana Frunza, Kashif Rasul, Yuriy Nevmyvaka, and Aidong Zhang. 2025. Chart-rvr: Reinforcement learning with verifiable rewards for explainable chart reasoning. *Preprint*, arXiv:2510.10973.

Bohao Tang, Yan Ma, Fei Zhang, Jiadi Su, Ethan Chern, Zhulin Hu, Zhixin Wang, Pengfei Liu, and Ya Zhang. 2025a. Visual programmability: A guide for code-as-thought in chart understanding. *Preprint*, arXiv:2509.09286.

Jiahao Tang, Henry Hengyuan Zhao, Lijian Wu, Yifei Tao, Dongxing Mao, Yang Wan, Jingru Tan, Min Zeng, Min Li, and Alex Jinpeng Wang. 2025b. From charts to code: A hierarchical benchmark for multimodal models. *Preprint*, arXiv:2510.17932.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Boran Wang, Xinming Wang, Yi Chen, Xiang Li, Jian Xu, Jing Yuan, and Chenglin Liu. 2025a. Chartagent: A chart understanding framework with tool integrated reasoning. *Preprint*, arXiv:2512.14040.

Fei Wang, Wenxuan Zhou, James Y. Huang, Nan Xu, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2024a. mdpo: Conditional preference optimization for multimodal large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 8078–8088. Association for Computational Linguistics.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, and 56 others. 2025b. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *Preprint*, arXiv:2508.18265.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, Alexis Chevalier, Sanjeev Arora, and Danqi Chen. 2024b. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Anran Wu, Luwei Xiao, Xingjiao Wu, Shuwen Yang, Junjie Xu, Zisong Zhuang, Nian Xie, Cheng Jin, and Liang He. 2023. Dcqa: Document-level chart question answering towards complex reasoning and common-sense understanding. *arXiv preprint arXiv:2310.18983*.

Shengguang Wu, Fan-Yun Sun, Kaiyue Wen, and Nick Haber. 2025. Symmetrical visual contrastive optimization: Aligning vision-language models with minimal contrastive images. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 30284–30297. Association for Computational Linguistics.

Renqiu Xia, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Botian Shi, Junchi Yan,

and Bo Zhang. 2025. Chartx and chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *IEEE Trans. Image Process.*, 34:7436–7447.

Luwei Xiao, Rui Mao, Shuai Zhao, Qika Lin, Yanhao Jia, Liang He, and Erik Cambria. 2025. Exploring cognitive and aesthetic causality for multimodal aspect-based sentiment analysis. *IEEE Transactions on Affective Computing*.

Zhengzhuo Xu, Sinan Du, Yiyan Qi, Chengjin Xu, Chun Yuan, and Jian Guo. 2024a. Chartbench: A benchmark for complex visual reasoning in charts. *Preprint*, arXiv:2312.15915.

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

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, Gongye Liu, Xiaomei Nie, Deng Cai, and Yujiu Yang. 2025a. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Yuwei Yang, Zeyu Zhang, Yunzhong Hou, Zhuowan Li, Gaowen Liu, Ali Payani, Yuan-Sen Ting, and Liang Zheng. 2025b. Effective training data synthesis for improving mllm chart understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2653–2663.

Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. 2024. Tinychart: Efficient chart understanding with program-of-thoughts learning and visual token merging. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 1882–1898. Association for Computational Linguistics.

Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2025. Chartcoder: Advancing multimodal large language model for chart-to-code generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 7333–7348. Association for Computational Linguistics.

Appendix

A Prompt Details

This section provides the detailed prompts used in this work.

A.1 Prompts for Counterfactual Data Synthesis

Due to the distinct nature of question types in existing datasets (Yang et al., 2025b; He et al., 2025), we design two separate prompts for code modification: one for descriptive questions and another for reasoning questions. This differentiation is motivated by the fact that reasoning questions, unlike descriptive ones, require the model to generate not only the final answer but also the intermediate reasoning steps.

- Figure 4: Prompt for descriptive questions.
- Figure 5: Prompt for reasoning questions, which includes instructions for generating intermediate reasoning steps.

A.2 Prompt for Image Similarity Evaluation

To implement our similarity-based data selection strategy (Section 3.3), we employ GPT-5-mini to evaluate the visual similarity between original and counterfactual chart pairs.

- Figure 6: Prompt for evaluating image similarity scores.

B Synthesis Cost and Data Validation

To provide a complete picture of ChartCF's data synthesis pipeline, we report the API cost, the multi-stage validation pipeline, and the end-to-end yield from seed samples to final training pairs.

API Cost. Table 8 summarizes the cost of each stage in our synthesis pipeline. The total cost for generating 10K counterfactual pairs is approximately $427, roughly 1/5 of the $2,145 reported by ECD (Yang et al., 2025b) for its full dataset. We note that this cost can be further reduced in several ways:

- (1) As shown in Figure 3, synthesizing only ∼4K pairs (without the filtering step) already matches ECD's performance on CharXiv, reducing the synthesis cost to approximately $162 (∼1/13 of ECD's cost).

Stage	Model	Cost (USD)
Counterfactual code generation (10K samples)	GPT-5	$403.93
Similarity scoring (10K pairs)	GPT-5-mini	$23.38
ChartCF synthesis total		**$427.31**

Table 8: API cost breakdown of ChartCF's data synthesis pipeline.

- (2) The relatively high cost is partly due to GPT-5's reasoning tokens. Replacing GPT-5 with GPT-4o as the code modifier yields comparable performance at substantially lower cost, as shown in Section 4.7.
- (3) Open-source models (e.g., Qwen3-VL-8B-Thinking) can also serve as effective code modifiers, allowing practitioners to run the code modification step locally and avoid API costs entirely.

Even accounting for the $71 cost of generating the 10K ECD seed samples we build upon, our full end-to-end pipeline totals $499 (∼1/4 of ECD's cost) while requiring significantly less training data (4K preference pairs vs. 300K SFT samples).

Data Validation. To ensure the quality of the synthesized counterfactual data, we adopt a multi-stage validation pipeline:

- **Stage 1: Feasibility check.** Our prompts (Figures 4 and 5) require the code modifier to first assess whether a meaningful modification is feasible before generating any code. Infeasible cases are filtered out at this stage. In practice, only 8 out of 10,512 seed samples (∼0.08%) are deemed infeasible.
- **Stage 2: Code parsing and execution.** The modified code is first parsed. Approximately 160 samples (∼1.5%) fail at this stage. Successfully parsed code is then executed to render the counterfactual chart image, with approximately 832 samples (∼8%) failing to render.
- **Stage 3: Retry mechanism.** For samples that fail at any of the above stages, we retry the code modifier up to two additional times. After retries, only 84 out of 10,512 samples (∼0.8%) ultimately fail to produce a valid counterfactual pair, yielding 10,428 valid pairs (99.2% success rate).
- **Stage 4: Human verification.** We further randomly sample 100 counterfactual pairs

generated by GPT-5 (50 descriptive and 50 reasoning) for manual inspection. For each pair, annotators check: (i) whether the code modification follows the prompt instructions and targets only answer-critical elements, (ii) whether the rendered counterfactual chart is visually plausible, and (iii) whether the new answer A_c is factually correct and can be reasonably inferred from the modified chart alone. Results show that 98 out of 100 pairs pass inspection (49/50 for descriptive and 49/50 for reasoning questions). While not perfect, this level of quality is sufficient for effective preference learning, as demonstrated by the performance improvements observed across five benchmarks in our experiments.

Finally, the similarity-based selection retains roughly 40% of the valid pairs (∼4K) for training. We also note that the similarity-based data selection strategy (Section 3.3) serves as an additional implicit quality filter, as poorly constructed counterfactual pairs with unintended large visual differences tend to be filtered out.

C Benchmark Details

We conduct experiments across five widely-adopted benchmarks, encompassing both real-world and synthetic charts.

Real-world Benchmarks. **CharXiv** (Wang et al., 2024b) contains 2,323 charts sourced from scientific literature, with 4,000 descriptive questions targeting basic chart element recognition and 1,000 reasoning questions requiring high-level reasoning across diverse chart elements. **ChartQA** (Masry et al., 2022) consists of 1,509 charts collected from 4 online sources and 2,500 question-answer pairs.

Synthetic Benchmarks. **ChartBench** (Xu et al., 2024a) provides 2,100 charts with 16,800 "yes/no" questions and 2,100 numerical questions. **ChartX** (Xia et al., 2025) contains 1,152 test samples across various chart types. **ECDBench** (Yang et al., 2025b) includes 1,224 chart images, each accompanied by one descriptive question and one reasoning question.

D Curriculum Learning for Hard Samples

To investigate whether curriculum learning could better leverage difficult samples, we design a two-stage training strategy. In the first stage, models are trained on low-similarity pairs (easier samples). In the second stage, we continue training with a reduced learning rate (1e-5) on either high-similarity pairs or data with distractors (see Section 4.6 for details about these hard sample types). Our motivation stems from the finding that directly training on overly difficult samples degrades performance. We hypothesize that starting with easier samples before progressively introducing harder ones might enable better handling of challenging cases.

Method	CharXiv		
	Des.	Rea.	Avg.
Qwen2.5-VL	66.40	41.20	61.36
Synthetic Data w/o Distractors			
ChartCF w/o Data Selection (10K)	73.08	43.20	67.10
w. Lowest Half Pairs (5K)	**75.65**	**43.70**	**69.26**
w. Highest Half Pairs (5K)	66.42	43.30	61.80
Curriculum Learning			
w. Stage-1: L (5K), Stage-2: H (5K)	75.60	42.80	69.04
w. Stage-1: L (5K), Stage-2: Dis (5K)	75.20	42.60	68.68

Table 9: Curriculum learning experiments on CharXiv. "L", "H", and "Dis" denote "Lowest Half Pairs", "Highest Half Pairs", and data with distractors, respectively.

Results in Table 9 show that curriculum learning fails to outperform the simpler strategy of using only low-similarity pairs. This suggests that in this setting, careful data selection is more effective than curriculum-based schedules. However, we believe there may exist more effective strategies to leverage such hard samples, a direction we plan to explore in future work.

E Case Study of Counterfactual Data Synthesis

We present four representative examples of counterfactual chart pairs generated by our synthesis pipeline, covering both descriptive and reasoning questions with varying modification types and similarity scores.

1. **Case 1** (Figure 7, similarity: 93) illustrates a descriptive question requiring legend interpretation. Our pipeline modifies the textual label in the legend from "Optimal Influence Zone" to "Critical Influence Zone" while keeping all visual elements unchanged, creating a relatively high-similarity counterfactual pair.

2. **Case 2** (Figure 8, similarity: 90) illustrates a descriptive question about axis scale recognition. The modification changes the maximum y-axis value from 250 to 200.

3. **Case 3** (Figure 9, similarity: 90) shows a reasoning question involving value extraction and numerical computation. Multiple data values are modified (e.g., Sun B Solar Flares from 6.8 to 9.7), resulting in a different final sum while maintaining similar chart appearance.

4. **Case 4** (Figure 10, similarity: 76) presents a reasoning question about spatial relationships between chart objects. The modification alters line positions, requiring identification of both visual elements and their relative positions.

These examples demonstrate our pipeline's ability to create diverse counterfactual pairs through different modification strategies: altering numerical values (Cases 2, 3), modifying textual information (Case 1), and changing visual element positions (Case 4).

F Case Study of Model Predictions

We compare predictions between ChartCF and the ECD baseline (both based on Qwen2.5-VL) on five challenging examples from CharXiv, demonstrating how our method enables more fine-grained understanding of chart details.

1. **Case 1** (Figure 11) involves a descriptive question requiring precise identification of a specific subplot and accurate reading of small-scale y-axis tick values. ECD fails to correctly identify the subplot location and misreads the axis range, while ChartCF correctly locates the target subplot and identifies the highest tick value as 1.5.

2. **Case 2** (Figure 12) presents a descriptive question about legend label extraction from a specific subplot. ECD incorrectly identifies the subplot and extracts labels from the wrong legend. ChartCF accurately locates the correct subplot and extracts only the relevant labels.

3. **Case 3** (Figure 13) demonstrates a reasoning question involving spatial position understanding. ECD misidentifies the target variable by confusing similar visual markers, while ChartCF correctly identifies "b_3" through precise spatial reasoning.

4. **Case 4** (Figure 14) shows a reasoning question requiring numerical computation from legend values. ECD makes calculation errors by misreading category counts, while ChartCF performs accurate value extraction and computation to reach the correct answer.

5. **Case 5** (Figure 15) illustrates a reasoning question about comparing visual patterns across subplot regions. ECD incorrectly interprets the slope comparison, while ChartCF accurately analyzes the curve steepness in different regions to determine the correct answer.

These cases highlight that ChartCF's counterfactual training enables models to develop more precise visual discrimination capabilities, particularly in tasks requiring fine-grained element localization, accurate value reading, spatial relationship understanding, and multi-step reasoning over chart elements.

Task: Given a chart image, its plotting code, a descriptive question, and the current answer, modify the code so that the answer to the question becomes different. You should ONLY modify the element(s) directly responsible for the current answer.

Requirements

- First assess whether you think you are capable of reasonably accomplishing this task
- Identify the specific data point(s) or element(s) that determine the current answer
- Modify ONLY those necessary elements to produce a different answer
- Do NOT change any other data points, labels, colors, or visual elements
- Do NOT change the final output/save path in the original code: it must remain 'rendered_images/{6-digit-number}.png', e.g., 'rendered_images/000002.png'.
- Do NOT modify the `set_random_seed` function or the random seed value it sets
- Ensure the modification is visually noticeable to human eyes (e.g., at least 15-25% change for numerical values)
- Provide the complete, executable Python code with your modifications, not just the changed parts

Example (Omitting the Chart Image for Brevity)

Example Input

Plotting Code:

```python
<example_original_code>
```

Question:
What is the title of the first subplot on the left?

Current Answer:
The title of the first subplot is 'Sculpture Wave Patterns'.

Example Output

Feasibility:
YES

Rationale of Modification:
To change the title of the first subplot, we only need to modify the `ax1.set_title()` function that sets the title of the first subplot. This change will directly affect the current answer without impacting any other part of the code or plot. Changing the title satisfies the requirement of producing a visually noticeable difference.

Modified Code:

```python
<example_modified_code>
```

New Answer:
The title of the first subplot is 'Dynamic Wave Effects'.

Input

Plotting Code:

```python
{{ python_code }}
```

Question:
{{ question }}

Current Answer:
{{ current_answer }}

Output Format

Feasibility:
[YES or NO - whether this task can be reasonably accomplished]

Rationale of Modification:
[If feasibility is YES: Briefly explain which element(s) you will modify and why this produces a different answer]
[If feasibility is NO: Briefly explain why]

Modified Code:

```python
[Your complete modified code here if feasible, otherwise write "None"]
```

New Answer:
[The new correct answer if feasible, otherwise write "None". Do NOT include words like "modified", "updated", "changed", or any reference to the modification process.]

Figure 4: The prompt used for descriptive questions.

Task: Given a chart image, its plotting code, a reasoning question, and the current answer with reasoning process, modify the code so that the answer becomes different. You should ONLY modify the element(s) directly responsible for the current answer.

Requirements

- First assess whether you think you are capable of reasonably accomplishing this task
- Identify the specific data point(s) or element(s) that determine the current answer
- Modify ONLY those necessary elements to produce a different answer with a reasoning process
- Do NOT change any other data points, labels, colors, or visual elements
- Do NOT change the final output/save path in the original code: it must remain 'rendered_images/{6-digit-number}.png', e.g., 'rendered_images/000002.png'.
- Do NOT modify the `set_random_seed` function or the random seed value it sets
- Ensure the modification is visually noticeable to human eyes (e.g., at least 15-25% change for numerical values)
- Provide the complete, executable Python code with your modifications, not just the changed parts

Example (Omitting the Chart Image for Brevity)

Example Input

Plotting Code:

```python
<example_original_code>
```

Question:
By how much does the mean revenue decrease from Q1 to Q2?

Current Answer:
Reasoning Process: The mean revenue for Q1 is 15.3 and for Q2 it is 11.9. The decrease is calculated as 15.3 - 11.9 = 3.4.
Answer: 3.4

Example Output

Feasibility:
YES

Rationale of Modification:
To change the answer, I will modify the mean revenue values for Q1 and/or Q2 in `revenue_means`. This adjustment will directly change the mean revenue values displayed in the chart without affecting other elements of the visualization.

Modified Code:

```python
<example_modified_code>
```

New Answer:
Reasoning Process: The mean revenue for Q1 is 19.1 and for Q2 it is 10.2. The decrease is calculated as 19.1 - 10.2 = 8.9.
Answer: 8.9

Input

Plotting Code:

```python
{{ python_code }}
```

Question:
{{ question }}

Current Answer:
Reasoning Process: {{ current_reasoning_process }}
Answer: {{ current_answer }}

Output Format

Feasibility:
[YES or NO - whether this task can be reasonably accomplished]

Rationale of Modification:
[If feasibility is YES: Briefly explain which element(s) you will modify and why this produces a different answer]
[If feasibility is NO: Briefly explain why]

Modified Code:

```python
[Your complete modified code here if feasible, otherwise write "None"]
```

New Answer:
Reasoning Process: [If feasible, provide step-by-step reasoning that leads to the new answer, Otherwise write "None". Do NOT include words like "modified", "updated", "changed", or any reference to the modification process.]
Answer: [The new correct answer if feasible, otherwise write "None". Do NOT include words like "modified", "updated", "changed", or any reference to the modification process.]

Figure 5: The prompt used for reasoning questions.

You are an expert at evaluating visualization chart plots. You will be given two python-generated chart images:
- **Original Image**: The chart before code modification
- **Modified Image**: The chart after code modification
Your task is to assess the similarity between the two chart images.

Scoring Criteria:
Evaluate the similarity between the two images based on the following criteria, totaling 100 points:

1. **Chart Types (20 points):** How similar are the chart types (e.g., line charts, bar charts, scatter plots, etc.) between the two images?
2. **Layout (20 points):** How similar is the arrangement of subplots (e.g., number of rows and columns, spacing) between the two images?
3. **Text Content (20 points):** How similar are the titles, annotations, axis labels, and other text elements (excluding axis tick labels) between the two images?
4. **Data (20 points):** How closely do the data trends, patterns, and the number of data groups match between the two images?
5. **Style (20 points):** How similar are the colors, line styles, marker types, legends, grids, and other stylistic details between the two images?

Evaluation:
Compare the two images head to head and provide a detailed assessment. Use the following format for your response:

—

Comments:
- Chart Types: {your comment and subscore}
- Layout: {your comment and subscore}
- Text Content: {your comment and subscore}
- Data: {your comment and subscore}
- Style: {your comment and subscore}

Score: {your final score out of 100}

—

Please use the above format to ensure the evaluation is clear and comprehensive.

Figure 6: The prompt used for evaluating image similarity scores.

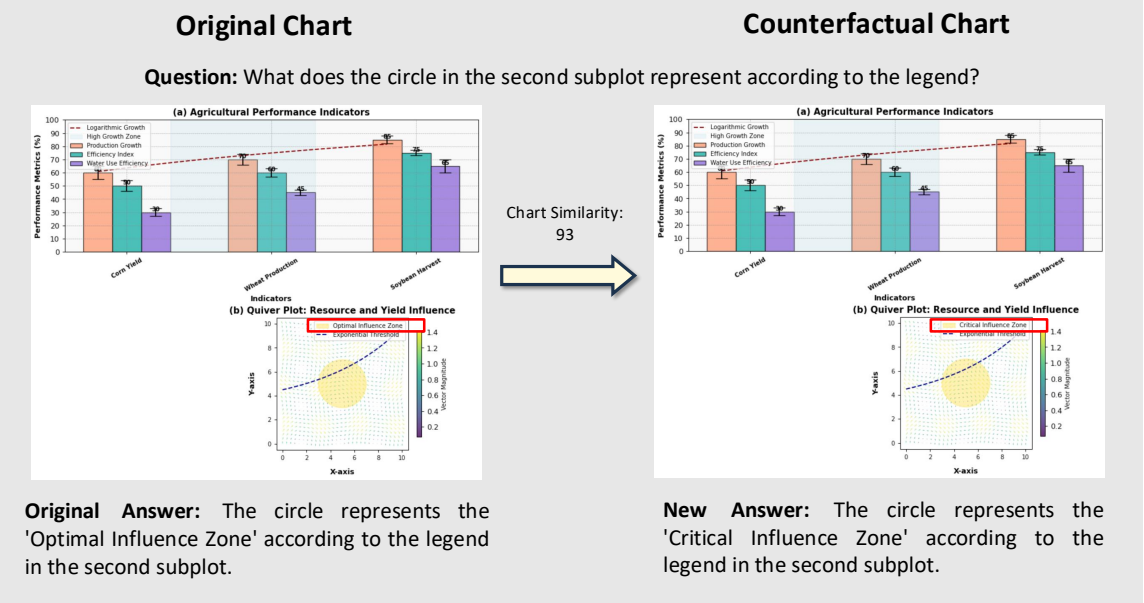


Figure 7: Counterfactual pair for a descriptive question about legend interpretation. The legend label “Optimal Influence Zone” is modified to “Critical Influence Zone” (similarity: 93).

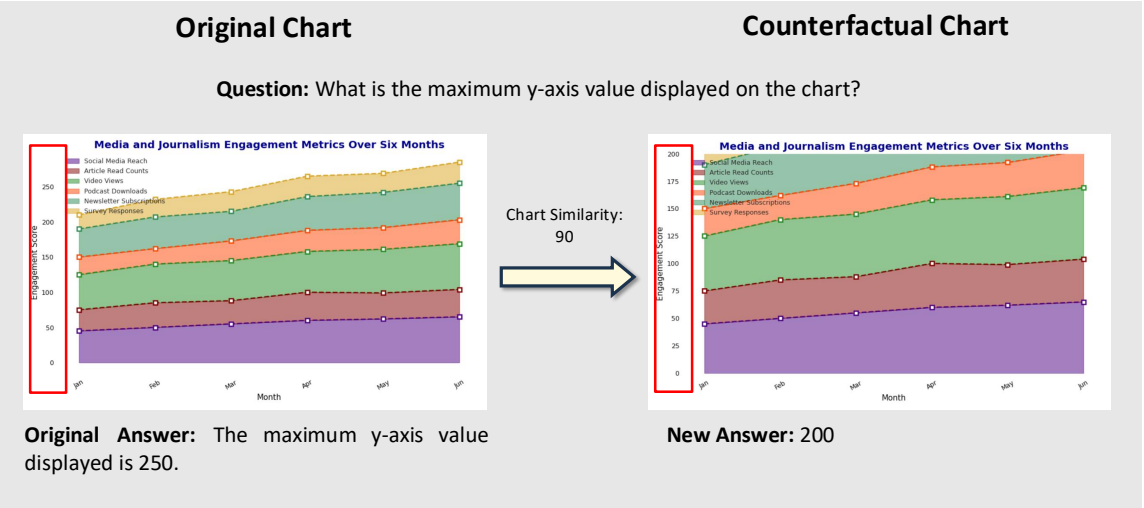


Figure 8: Counterfactual pair for a descriptive question about y-axis scale. The maximum y-axis value is changed from 250 to 200 (similarity: 90).

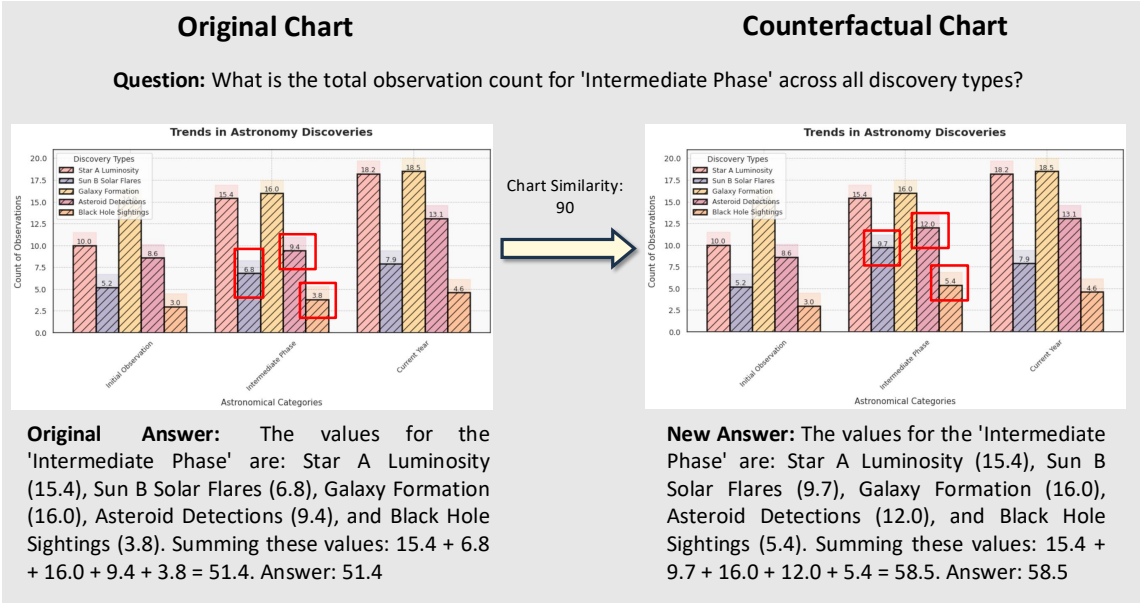


Figure 9: Counterfactual pair for a reasoning question involving numerical computation. Multiple data values are modified, resulting in a different sum (similarity: 90).

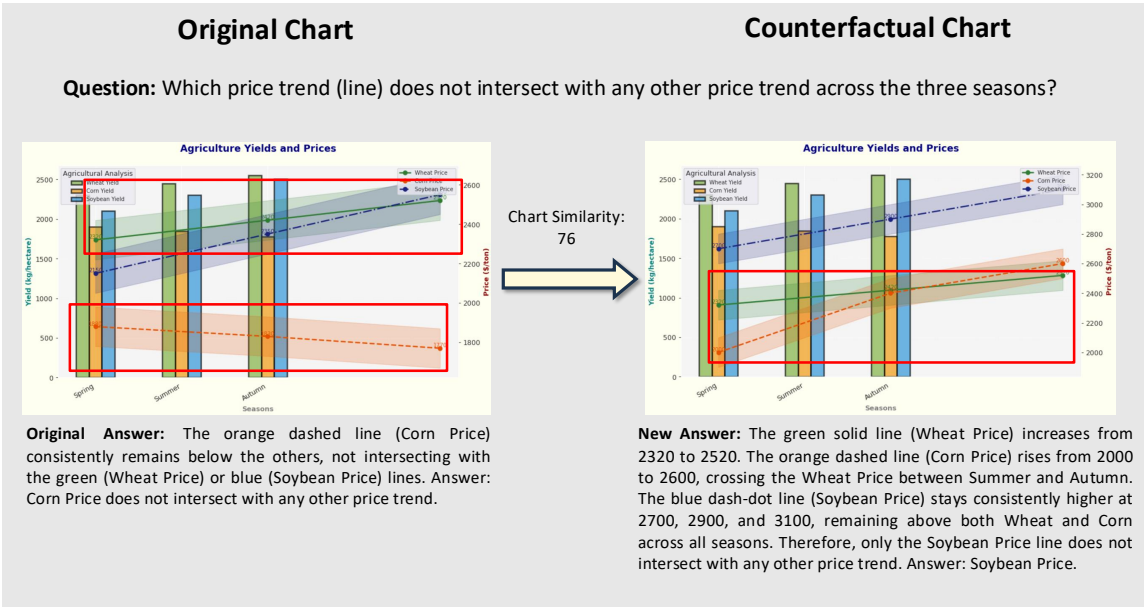


Figure 10: Counterfactual pair for a reasoning question about line intersections. The Corn Price trend is modified from declining to increasing, changing spatial relationships (similarity: 76).

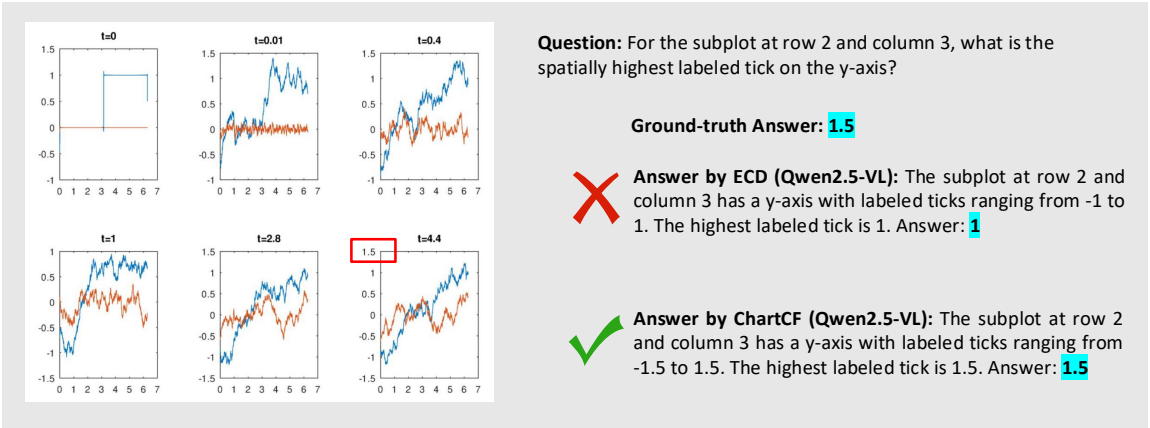
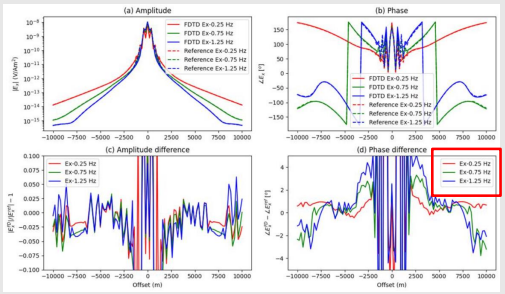


Figure 11: Comparison on a descriptive question about y-axis tick values. ECD misidentifies the subplot location and axis range, while ChartCF correctly identifies the target subplot and reads the highest tick value.



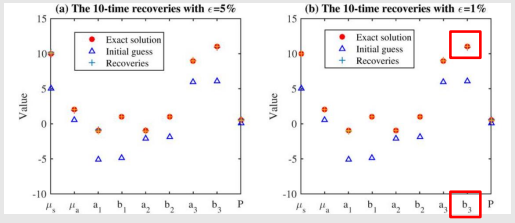
Question: For the subplot at row 2 and column 2, what are the names of the labels in the legend?

Ground-truth Answer: **Ex-0.25 Hz, Ex-0.75 Hz, Ex-1.25 Hz**

Answer by ECD (Qwen2.5-VL): The legend in subplot (b) contains the following labels: 'FDTD Ex-0.25 Hz', 'FDTD Ex-0.75 Hz', 'FDTD Ex-1.25 Hz', 'Reference Ex-0.25 Hz', 'Reference Ex-0.75 Hz', and 'Reference Ex-1.25 Hz'. Answer: **FDTD Ex-0.25 Hz, FDTD Ex-0.75 Hz, FDTD Ex-1.25 Hz, Reference Ex-0.25 Hz, Reference Ex-0.75 Hz, Reference Ex**

Answer by ChartCF (Qwen2.5-VL): The legend in subplot (d) contains the following labels: 'Ex-0.25 Hz', 'Ex-0.75 Hz', 'Ex-1.25 Hz'. These labels correspond to the curves in the plot. Answer: **Ex-0.25 Hz, Ex-0.75 Hz, Ex-1.25 Hz**

Figure 12: Comparison on a descriptive question about legend labels. ECD extracts labels from the wrong subplot, while ChartCF accurately locates the correct subplot and extracts the relevant labels.



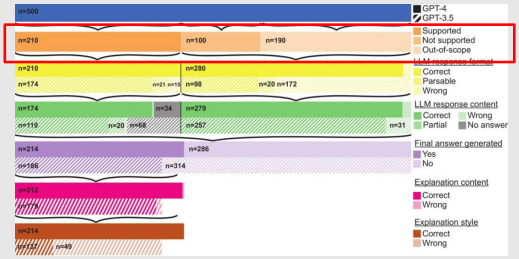
Question: Which variable in subplot (b) has the highest exact solution?

Ground-truth Answer: **b_3**

Answer by ECD (Qwen2.5-VL): The highest exact solution value in subplot (b) corresponds to 'P', which reaches approximately 10 on the y-axis. Answer: **P**

Answer by ChartCF (Qwen2.5-VL): The highest exact solution value in subplot (b) corresponds to 'b_3', which reaches approximately 10 on the y-axis. Answer: **b_3**

Figure 13: Comparison on a reasoning question about identifying the highest solution value. ECD confuses similar visual markers, while ChartCF correctly identifies “b_3” through precise spatial reasoning.



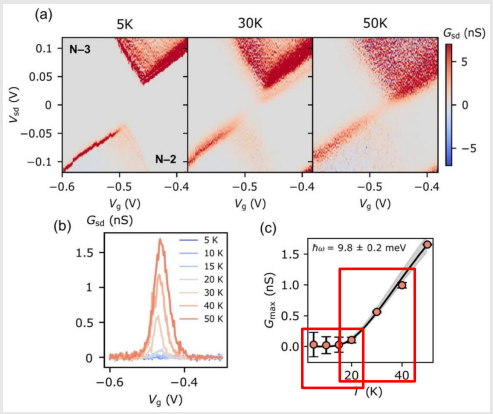
Question: How many less instances were there of supported cases compared to the sum of not supported and out-of-scope cases?

Ground-truth Answer: **80**

Answer by ECD (Qwen2.5-VL): The number of supported cases is 210 (GPT-4) + 100 (GPT-3.5) = 310. The sum of not supported and out-of-scope cases is 174 + 190 = 364. The difference is 364 - 310 = 54. Answer: **54**

Answer by ChartCF (Qwen2.5-VL): The total number of not supported and out-of-scope cases is 100 + 190 = 290. The number of supported cases is 210. The difference is calculated as 290 - 210 = 80. Answer: **80**

Figure 14: Comparison on a reasoning question involving numerical computation from legend categories. ECD makes calculation errors due to misreading counts, while ChartCF performs accurate value extraction and computation.



Question: Compare the value increase between 0-20 and 20-40 in subplot (c). Which one is larger?

Ground-truth Answer: **20-40**

Answer by ECD (Qwen2.5-VL): The slope of the curve from 0-20 K is steeper than that from 20-40 K, indicating a larger relative increase in Gmax over this range. Answer: **The increase from 0-20 K is larger.**

Answer by ChartCF (Qwen2.5-VL): The slope of the curve from 0-20 (nearly horizontal) is smaller compared to the slope from 20-40 (steeply increasing). Answer: **The increase from 20-40 is larger.**

Figure 15: Comparison on a reasoning question about comparing curve slopes across regions. ECD incorrectly interprets the slope comparison direction, while ChartCF accurately analyzes curve steepness to determine the larger increase.