ChartAlign: Instance-Level Visual Alignment for Robust Chart Understanding in MLLMs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Despite significant advances in Multimodal Large Language Models (MLLMs) for standard chart understanding, existing models experience significant performance degradation when presented with semantically equivalent variants of the standard chart, such as standard charts without explicit textual annotations or pictorial charts with complex visual elements. This suggests that existing MLLMs rely more heavily on textual cues and conventional shapes rather than robust visual comprehension. To address this issue, we first introduce **ChartPairs**, a novel dataset consisting of pairs of standard charts and their visually diverse yet semantically equivalent variants. Leveraging this dataset, we propose **ChartAlign**, a novel instance-level alignment method for image encoders that can be seamlessly integrated into existing models without requiring full retraining. Compared to traditional distribution-level alignment methods, ChartAlign ensures theoretically stronger visual consistency across equivalent charts. Extensive experiments across multiple chart-related tasks demonstrate that MLLMs enhanced with ChartAlign significantly outperform state-of-the-art baselines on challenging variants.

1 Introduction

Charts play a vital role in visualizing complex data and facilitating effective communication in various domains, including scientific research [1], decision making [2], and emotional communication[3]. Recently, Multimodal Large Language Models (MLLMs) have demonstrated remarkable capabilities in understanding and interpreting standard charts, bridging the gap between visual data representations and natural language understanding [4, 5, 6, 7, 8].

Despite these advancements, existing models often struggle with visually diverse yet semantically equivalent charts. For example, Wu *et al*. [9] reports a 23.8% average performance drop when textual annotations are removed. Our preliminary experiment results also showed a drop of 22.06% among different models when facing artistically stylized charts compared to standard charts. These findings suggest that existing MLLMs, especially their image encoders, rely heavily on explicit textual annotations and standard visual patterns when recognizing and understanding charts. This limits their robustness and generalization capabilities for diverse real-world charts.

To address these challenges, it is necessary to adapt the models with more diverse charts. However, joint finetuning of both vision and language components incurs high computational costs and requires extensive data, while language-only tuning neglects critical visual feature extraction capabilities required to understand visually diverse charts. Therefore, there is an urgent need for more effective and efficient adaptation methods tailored explicitly to enhancing the image encoders' ability to generalize across visually diverse charts.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

To support this, we first introduce **ChartPairs**, a carefully constructed dataset containing pairs of standard charts and their semantically equivalent counterparts with diverse visual styles. Leveraging this dataset, we propose **ChartAlign**, a novel instance-level alignment method specifically designed to improve the robustness of image encoders within MLLMs. Unlike traditional distribution-level alignment methods, which may lead to imprecise alignment and negative transfer (*i.e.*, degraded performance due to misaligned representations), our instance-level method directly aligns feature representations of paired charts. This ensures stronger visual consistency and more precise semantic preservation across diverse visual representations, enhancing models' generalization capabilities without requiring costly full retraining. Furthermore, this alignment method can be readily integrated into existing MLLMs without requiring full retraining, offering a plug-and-play solution compatible with diverse chart-related tasks. The evaluation results demonstrate the effectiveness of our method in enhancing generalization to visually diverse charts across multiple tasks and baselines.

The primary contributions of our work are:

- We identify and formulate the core challenge that existing MLLMs heavily rely on textual annotations and standard shapes, limiting their generalization to visually diverse charts.
- We construct and release **ChartPairs**, a novel dataset enabling effective instance-level alignment across diverse charts.
- We propose **ChartAlign**, a novel instance-level alignment strategy specifically designed to enhance the visual comprehension capability of image encoders.

2 Related Work

2.1 Chart related MLLMs

Multimodal large language models (MLLMs) utilize connectors to bridge large language models [10, 11, 12, 13] and vision encoders [14, 15], enabling enhanced comprehension and instruction-following capabilities. Methods such as BLIP2 [16], Flamingo [17], and Qwen-VL [18] employ QFormers or Resamplers to align modalities using extensive datasets of image-text pairs. LLaVA [19, 20] pioneered the extension of instruction tuning to visual tasks, achieving impressive performance with a simple MLP that preserves visual information while refining multimodal alignment. LLaVA-HR [21] introduces a Mixture-of-Resolution Adaptation (MRA) framework to enhance the visual understanding of MLLMs by adapting to different chart resolutions.

In the domain of chart understanding, MLLMs have been adapted through various architectural innovations. Early approaches like Pix2Struct [22] and MatCha [23] focus on aligning chart content with alternative representations such as markdown or tables. DePlot [24] employs a two-stage approach by fine-tuning models for table extraction before leveraging LLMs for reasoning, while ChartVLM [25] incorporates a discriminator to determine when LLM intervention is necessary. Moving toward more integrated solutions, models such as ChartLlama [26] build upon LLaVA's foundation to incorporate diverse chart types and downstream tasks. ChartPaLI [27], ChartAst [28], and MMC [29] focus on table-chart alignment. OneChart [8] and ChartMoE [6] align charts with structured formats like JSON and Python dictionaries, while ChartMoE utilizes Mixture of Experts (MoE) to handle the complexity of chart understanding. To address the challenge of processing high-resolution charts efficiently, TinyChart [5] employ token merging strategies that preserve visual fidelity while reducing computational demands.

2.2 Limitations in Chart Verification

Recent studies highlight critical gaps in chart verification systems. While existing methods leveraging OCR and LLMs [30, 31] demonstrate basic fact-checking capabilities, they frequently miss visual manipulations like axis truncations or distorted scales due to overreliance on extracted numerical data. Even vision-language models [25, 26] exhibit limited sensitivity to visual-data inconsistencies, as they prioritize textual/numerical information over graphical semantics [9]. This oversight enables malicious actors to craft misleading charts with surface-level data plausibility [32, 33], revealing the need for frameworks that jointly analyze visual encodings, statistical relationships, and contextual claims.

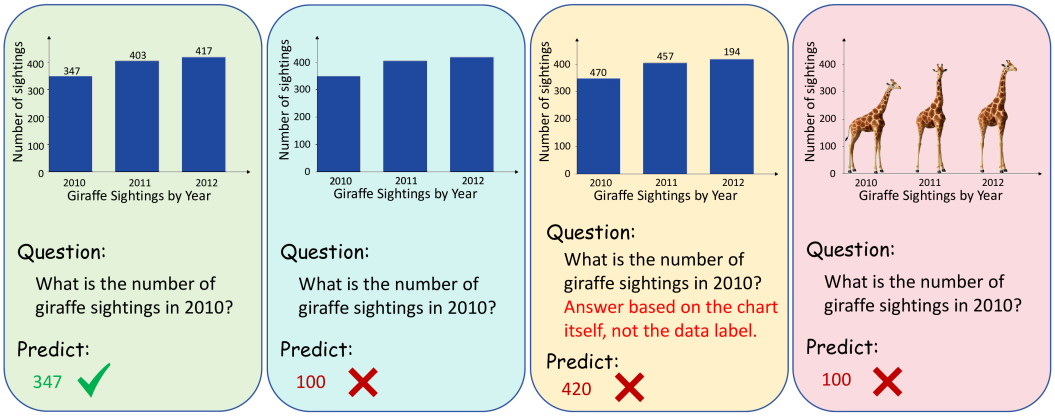

Figure 1: Illustrative examples showing ChartLLaMa's responses across chart variants. The standard chart (green) elicits accurate responses, while the standard chart without text (blue), the standard chart with misleading text (yellow), and the pictorial charts without data labels (pink) yield incorrect value estimations.

2.3 Discrepancy-based domain adaptation

Domain Adaptation (DA) enables models trained on a source domain to be effectively transferred to a target domain by minimizing the discrepancy between domains. Based on the type of domain divergence (distribution shift or feature space difference), DA can be categorized into homogeneous and heterogeneous approaches [34].

Various methods have been proposed to achieve effective domain adaptation. For instance, [35, 36] introduced the Maximum Mean Discrepancy (MMD) loss to minimize feature distribution differences by computing the norm between domain means. Additionally, some approaches focus on optimizing network architecture. [37] proposed using weight regularizers to relate corresponding layer weights across domains, while [38] employed weakly parameter-shared layers. These methods have demonstrated effectiveness in both supervised and unsupervised settings.

3 Preliminary Analysis: Revisiting MLLMs for Different Chart Variants

As shown in Figure 1, we evaluated ChartLLaMa [26] using a standard chart (green) and three different variants. The results reveal critical limitations that the model performs accurately with standard charts but fails to estimate values correctly, indicating overreliance on standard visual patterns and explicit labels. Despite explicit instructions to rely solely on visual data, the model consistently prioritizes misleading textual labels over contradicting visual evidence. The quantitative results in Table 1 further confirm a performance drop of 25.12% and 37.36% when presented with standard charts and pictorial charts without textual labels. These findings highlight the urgent need to enhance MLLMs' visual encoding capabilities to better interpret chart semantics across diverse visual styles while reducing dependence on textual annotations.

4 Method

4.1 Overview

To overcome the limitations of MLLMs in understanding visually diverse charts, we propose a novel framework combining **ChartPairs**, a dataset of semantically equivalent chart pairs, and **ChartAlign**, an instance-level alignment method for image encoders. As depicted in Figure 2, ChartAlign utilizes a teacher encoder (f_t) from a pre-trained model and a student encoder (f_s) initialized with the teacher's parameters. The student encoder is then optimized to align feature representations of paired charts within ChartPairs. This framework enhances model visual comprehension capability and robustness by enabling image encoders to generalize across charts with varied visual styles, such as text-free or pictorial variants, without requiring costly full-model retraining.

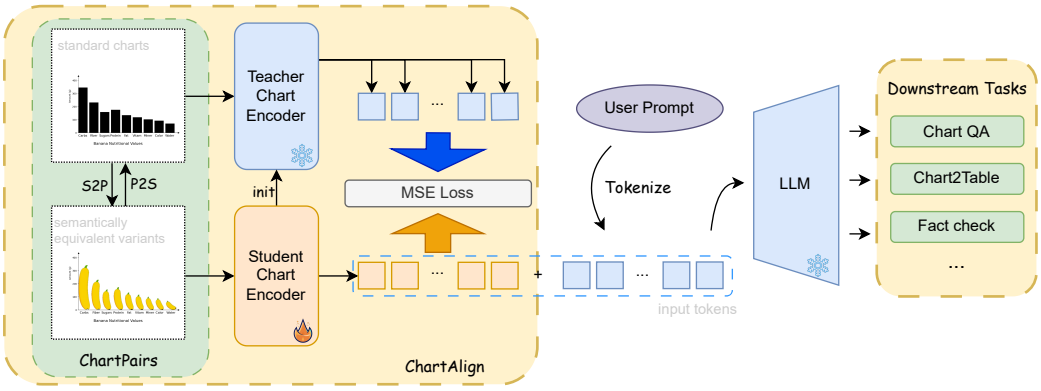

Figure 2: The overview of our proposed ChartAlign framework. Left: The frozen teacher encoder guides the student encoder to learn and align features. Right: The aligned features are directly fed into downstream models, enabling support for diverse tasks.

4.2 ChartPairs

4.2.1 Chart Variants

Following the preliminary analysis, our study focuses on the following four chart variants: standard charts with labels ($C^{\text{s,l}}$), standard charts without labels ($C^{\text{s,n}}$), pictorial charts with labels ($C^{\text{p,l}}$), and pictorial charts without labels ($C^{\text{p,n}}$). Note that our framework can be easily extended to include more variants.

4.2.2 Chart Generation

The key to constructing ChartPairs is to generate semantically equivalent chart pairs with different visual styles. To achieve this, we simultaneously employ two pipelines: a Pictorial-to-Standard (P2S) pipeline that first constructs pictorial charts through generative AI and then extracts the equivalent standard charts, and a Standard-to-Pictorial (S2P) pipeline that first renders standard charts and then transforms them into pictorial ones. This dual-pipeline architecture addresses fundamental limitations inherent to each individual one: the P2S method excels at producing visually coherent and aesthetically pleasing charts but struggles with precise data fidelity, while the S2P method offers perfect data fidelity but may lack aesthetics and creativity. By developing both pipelines in parallel, ChartPairs contains more diverse charts and also provides users with flexibility to prioritize either of them if they have specific needs in their applications.

Pictorial-to-Standard (P2S) Pipeline. The P2S pipeline comprises three stages: (1) diffusion-based pictorial charts generation, (2) salient visual component segmentation, and (3) chart variants generation. Specifically, it first utilizes Flux [39] to generate initial pictorial chart prototypes This method, despite its visual coherence and aesthetics, usually exhibits two critical artifacts: inconsistent textual annotations and visual element overlap. To resolve these issues, we introduce a language-driven semantic segmentation method to extract visual components and remove textual labels and axes, where DINO [40] performs semantic component detection and SAM-2 [41] executes pixel-precise segmentation. Based on the segmentation results, we calculate the bounding box of each element and reconstruct the original data, which allows us to generate the four chart variants by adding the correct axes and optional textual annotations to processed pictorial charts and standard charts.

Standard-to-Pictorial (S2P) Pipeline. Despite the P2S pipeline's ability to generate visually appealing charts, it sometimes fails to accurately represent data values due to segmentation errors, which may lead to potential misinterpretations. To overcome this, we also developed an S2P pipeline that ensures precise data fidelity by first rendering standard charts and then converting them into pictorial ones. Since the generation of standard charts with and without textual annotations is straightforward, we focus on the pictorialization process, which consists of the following two stages:

Retrieve Relevant Visual Elements.

4.2.3 Paired Chart Construction

Following the chart generation, we explain how we construct paired charts to enhance model capability. Specifically, for each ordered pair $(X^{\text{source}}, X^{\text{target}})$, the teacher encoder processes the source chart (X^{source}) to extract features that guide the student encoder in learning to process the target chart (X^{target}). In this work, we construct paired charts to handle two key scenarios: label-agnostic chart understanding and pictorial chart understanding.

Label-agnostic Chart Understanding. To train models capable of analyzing charts without relying on textual annotations, we create pairs where standard charts without labels ($C^{\text{s,n}}$) serve as the source, paired with each of the other three chart variants ($C^{\text{s,l}}, C^{\text{p,n}}, C^{\text{p,l}}$) as targets:

$$\mathcal{D}_{\text{label-agnostic}} = \{(C_i^{\text{s,n}}, C_i^{\text{s,l}}), (C_i^{\text{s,n}}, C_i^{\text{p,n}}), (C_i^{\text{s,n}}, C_i^{\text{p,l}})\}_{i=1}^{N}. \tag{1}$$

Finetuning image encoders on this dataset encourages models to disregard data labels and pictorial elements, focusing solely on the underlying visual representation of data. This is particularly important because it allows models to identify misleading or manipulated charts by relying on the pure visual structure, independent of potentially inaccurate or deceptive textual annotations.

Pictorial Chart Understanding. To enable MLLMs to understand pictorial charts, we create pairs where standard charts ($C^{\text{s,n}}, C^{\text{s,l}}$) serve as the source and pictorial charts ($C^{\text{p,n}}, C^{\text{p,l}}$) as the target:

$$\mathcal{D}_{\text{pictorial}} = \{(C_i^{\text{s,n}}, C_i^{\text{p,n}}), (C_i^{\text{s,l}}, C_i^{\text{p,l}})\}_{i=1}^{N}. \tag{2}$$

Note that within each pair, the source standard chart and the target pictorial chart either both include data labels or both exclude them. This design is made to better leverage the capability of the original model in understanding charts with textual annotations. In other words, all such pairs share identical underlying data distributions but differ solely in their visual presentation styles.

4.3 ChartAlign

After constructing paired charts, we train the student encoder to look beyond visual differences and recognize the same underlying data. The objective function is to align the features of target images produced by the student encoder with the features of source images produced by the teacher encoder. The total loss function $\mathcal{L}_{\text{total}}$ is hence defined as:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{align}} + \lambda_2 \mathcal{L}_{\text{consistent}}, \tag{3}$$

where $\mathcal{L}_{align}$ is the feature alignment loss applied to the source-target pairs $(x_i^{\text{source}}, x_i^{\text{target}})$:

$$\mathcal{L}_{\text{align}} = \frac{1}{N} \sum_{i=1}^{N} \| f_s(x_i^{\text{target}}) - f_t(x_i^{\text{source}}) \|_2^2, \tag{4}$$

where f_s, f_t are the student and teacher encoder, respectively, and $\mathcal{L}_{consist}$ is the consistency loss only applied to the source images:

$$\mathcal{L}_{\text{consistent}} = \frac{1}{N} \sum_{i=1}^{N} \| f_s(x_i^{\text{source}}) - f_t(x_i^{\text{source}}) \|_2^2. \tag{5}$$

This consistency term serves two crucial purposes. First, it maintains stable representations for standard charts to preserve downstream task performance because those parameters are frozen. Second, it helps prevent the student encoder from generating collapsed feature representations that might artificially inflate alignment scores.

Here, λ_1 and λ_2 are weights that balance feature alignment and consistency preservation. To effectively balance these two terms, we employ Dynamic Weight Averaging (DWA) [42], which dynamically adjusts weights based on the difficulty of each task. Specifically, we first compute the loss ratio between consecutive epochs for each task $w_{k,t-1} = \mathcal{L}_{k,t-1}/\mathcal{L}_{k,t-2}$, and then derive the weights by applying an exponential mapping normalized across tasks $\lambda_{k,t} = \exp(w_{k,t-1}/T)/\sum_i \exp(w_{i,t-1}/T)$, where T is a temperature parameter controlling task coupling and set as 2 in our implementation.

4.4 Justification for Instance-Level Alignment

Traditional domain adaptation techniques often focus on aligning the marginal feature distributions between the source and target domains. This is typically achieved by optimizing an objective such as: $\min_\theta \mathcal{L}_{\text{task}}(X^{\text{source}}; \theta) + \lambda \cdot D(P(f_\theta(X^{\text{source}})), P(f_\theta(X^{\text{target}})))$, where f_θ is a feature extractor, and $D(\cdot, \cdot)$ is a divergence measure that quantifies the difference between source feature distribution and target feature distribution. Our proposed method takes a more direct path by constructing explicit pairs of source and target instances $(x_i^{\text{source}}, x_i^{\text{target}})$ and minimizing the distance between their respective feature representations, which brings two benefits over distribution-level alignment.

First, while the distribution-level alignment ensures that the distribution $P(f_\theta(X^{\text{source}})) \approx P(f_\theta(X^{\text{target}}))$, it does not guarantee the alignment of the crucial conditional distributions $P(Y \mid f_\theta(X^{\text{source}}))$ and $P(Y \mid f_\theta(X^{\text{target}}))$. Therefore, the model learned on the source domain may not be readily applicable to the target domain.

Second, it avoids negative transfer for paired instances. Traditional distribution-level alignment will potentially produce erroneous alignments where charts are close in a feature projection but are semantically disparate. By focusing on specific, pre-defined pairs believed to be semantically equivalent, our method ensures that these particular cross-domain counterparts are mapped close together in the learned feature space.

5 Experiments

5.1 Implementation Details

During the alignment process, we freeze the teacher vision encoder and only update the student vision encoder. At inference time, the optimized student vision encoder is employed for feature extraction to feed the downstream language model. All training processes are done on 4 $\times$ GTX 4090 GPUs in less than 5 hours. Refer to the Appendix 8.1 for more details.

5.2 Dataset

Following the pipeline introduced in Section 4.2.2, we developed ChartPairs, which contains 4,616 charts across four primary chart types: bar chart (2716), pie chart (900), line chart (500), and scatter plot (500). To ensure unbiased evaluation, we split ChartPairs into non-overlapping training (3,922 charts) and test (694 charts) sets. Each chart includes variations based on two factors: visual presentation style (standard, pictorial) and data label configuration (correct labels, no labels).

To evaluate model performance on downstream tasks, we also established the ground truth results for the different chart-related tasks. Our evaluation focuses on two mainstream tasks, Chart2Table and ChartQA. The ground truth for Chart2Table is the original data table, which can be naturally obtained during the data construction process. For ChartQA tasks, we carefully designed question-answer pairs targeting different aspects of chart comprehension, including numerical value extraction, extreme value identification, and distribution analysis. The questions are generated through a two-stage process: we first create template-based questions using rule-based algorithms based on chart metadata, and then refine the description with GPT-4o-mini to improve linguistic variety while preserving evaluation objectives. The detailed statistics of the dataset, all templates and prompts employed in the construction process, and representative examples are presented in the supplemental material.

5.3 Baseline Methods

To comprehensively evaluate our method's generalizability, we consider both general-purpose MLLMs and specialized chart understanding models in our evaluation. For general-purpose models, we incorporated three LLaVa variants (LLaVa-HR-7b, LLaVa-HR-13b [21], and LLaVa1.6-13b [43]) because they are representative and widely adopted multimodal architectures. We also include Qwen2.5-VL-7b [18], which demonstrates state-of-the-art performance on various vision-language benchmarks. For specialized chart understanding models, we include Matcha [23], which implements a highly efficient, lightweight architecture specifically optimized for chart interpretation tasks. We further included four language model-based models: ChartLLaMa [26], TinyChart [5], ChartInstruct [7],

Model	Param.	Pictorial Chart			Standard Chart			Overall avg.
		w/ dl	w/o dl	avg.	w/ dl	w/o dl	avg.	
General-purpose Multi-modal Large Language Models								
LLaVa-HR-7b	0.5b+6.8b	26.34	11.07	18.71	27.67	11.50	19.59	19.15
LLaVa-HR-13b	1.2b+13b	32.57	11.34	21.96	32.78	12.40	22.59	22.28
LLaVa1.6-13b	0.3b+13b	52.05	23.90	37.98	58.54	39.12	48.83	43.41
Qwen2.5-VL-7b	0.7b+7.6b	88.29	45.98	67.14	89.78	**78.77**	84.28	75.71
Qwen2.5-VL-7b (+Ours)	0.7b+7.6b	**90.74**	**76.42**	**83.58**	**91.06**	78.13	**84.60**	**84.09**
Specialized Chart Understanding Models								
ChartLLaMa	0.3b+13b	44.33	15.81	30.07	53.17	28.05	40.61	35.34
Matcha	92m+190m	32.84	30.12	31.48	44.70	42.84	43.77	37.63
Matcha (+Ours)	92m+190m	44.81	41.03	42.92	45.29	41.94	43.62	43.27
TinyChart	0.4b+2.8b	54.34	27.67	41.01	**74.77**	60.99	67.88	54.45
TinyChart (+Ours)	0.4b+2.8b	54.12	27.20	40.66	**74.77**	61.15	67.96	54.31
ChartInstruct	74m+6.8b	53.49	26.77	40.13	67.54	57.74	62.64	51.34
ChartInstruct (+Ours)	74m+6.8b	66.74	51.30	59.02	66.90	55.56	61.23	60.13
ChartMLLM	1.2b+13b	61.31	28.26	44.79	70.52	62.96	66.74	55.77
ChartMLLM (+Ours)	1.2b+13b	**70.57**	**63.70**	**67.14**	70.46	**64.66**	**67.56**	**67.35**

Table 1: Performance comparison of models in the ChartQA task. Results are reported for both Pictorial and Standard charts with and without data labels. Param. indicates the parameter count of vision encoder and other components (mainly from LLM + few from MLP connector).

and ChartMLLM [4]. These models demonstrate superior performance on chart-related downstream tasks after training on various charts.

5.4 Evaluation Metrics

We evaluate model performance using established metrics for ChartQA task and Chart2Table task.

ChartQA Metric. For the ChartQA task, we employ Relaxed Accuracy as our primary evaluation metric following [44, 45]. Non-numeric answers use exact string matching after conversion to lowercase. Percentage answers are standardized to a 0-100 scale. Numerical answers are considered correct if it is within 10% of the gold answer, *i.e.*, $|y - \hat{y}|/|y| \leq 0.1$.

Chart2Table Metric. For the Chart2Table task, we adopt the Relative Mapping Similarity (RMS) proposed in DePlot [24]. RMS extracts a similarity matrix between predicted and ground truth tables identifying minimal cost matching, which evaluates how effectively the model extracts the underlying data considering both cell values and structural alignment. Afterwards, we compute F1 score according to the similarity matrix following DePlot [24].

5.5 Results

ChartQA Task. Table 1 compares ChartAlign against various ChartQA models. For general-purpose MLLMs, LLaVa variants show limited chart understanding capability (avg: 19.59%-48.83%), with severe performance degradation on pictorial charts without labels (11.07%-23.90%). While Qwen2.5-VL-7b achieves substantially better performance (avg: 75.71%), but still exhibits considerable weakness on pictorial charts without labels (45.98%). Specialized chart models show better performance than LLaVa variants with fewer parameters, highlighting the benefits of domain-specific training. However, they also struggle with pictorial charts and standard charts without labels. After incorporating with ChartAlign, most baseline methods exhibit significant improvement on pictorial chart understanding and maintain comparable performance on standard charts. Even the best performing model, Qwen2.5-VL-7b, shows meaningful gains: slight improvement on standard charts (84.28% $\rightarrow$ 84.60%) and substantial improvement on pictorial charts (67.14% $\rightarrow$ 83.58%), especially on pictorial charts without labels (45.98% $\rightarrow$ 76.42%). Similar phenomena can also be observed when applying ChartAlign into specialized chart understanding models. ChartMLLM, the top performer among specialized models, exhibits significant performance gains in pictorial chart understanding while maintaining stable performance on standard charts when augmented with our method. The only exception is TinyChart [5], which uses a dynamic token merging policy in the vision encoder that disrupts the visual token matching relationships, making it incompatible with our method.

Model	Weighting	Pictorial Chart			Standard Chart		
		w/ dl	w/o dl	ave	w/ dl	w/o dl	ave
ChartMLLM	-	61.31	28.26	44.79	70.52	62.96	66.74
ChartMLLM (+Ours)	fixed	70.57	63.70	67.14	70.46	64.66	67.56
ChartMLLM (+Ours)	DWA	69.77	61.69	65.73	70.46	63.23	66.85
Qwen2.5-VL-7b	-	88.29	45.98	67.14	89.78	78.77	84.28
Qwen2.5-VL-7b (+Ours)	fixed	**90.90**	76.10	83.50	90.89	77.96	84.43
Qwen2.5-VL-7b (+Ours)	DWA	90.74	**76.42**	**83.58**	**91.06**	**78.13**	**84.60**

Table 4: Performance comparison of varying multi-task weighting strategies. "fixed" denotes a fixed weight $(\lambda_1, \lambda_2) = (1, 1)$ and "DWA" denotes dynamic weight averaging.

Chart2Table Task. ChartAlign also significantly enhances the Chart2Table extraction capabilities. Here we compare ChartMLLM, which demonstrates state-of-the-art performance on this task. As shown in Table 2, our method improves ChartMLLM's performance on pictorial charts without labels from 20.78% to 79.82%, and pictorial charts with labels from 68.13% to 87.92%. This improvement enables accurate data extraction despite complex visual presentations.

Model	Pictorial		Standard	
	w/ dl	w/o dl	w/ dl	w/o dl
ChartMLLM	68.13	20.78	79.06	69.63
ChartMLLM (+Ours)	87.92	79.82	82.41	73.50

Table 2: Performance comparison of models in the Chart2Table task.

Misleading Charts Understanding. To further evaluate the model's capability to focus on interpreting visual elements when instructed to ignore labels, we test ChartQA performance on charts with misleading labels, which are constructed by manipulating the text labels with a random number. Specifically, we add additional prompts in this task to guide the model to answer questions focusing on chart representations rather than data labels. The image encoders are fine-tuned using $\mathcal{D}_{\text{label-agnostic}}$ to steer the model's focus on visual elements rather than textual annotation. Table 3 presents the results tested on both pictorial and standard charts with misleading data labels. All the models with ChartAlign achieves similar results on misleading labels compared with the their base models on standard chart without data label (*e.g.*, 61.31% vs. 62.96% on ChartMLLM). This consistency also proves that our ChartAlign of $\mathcal{D}_{\text{label-agnostic}}$ can regard charts with data label as those without.

Model	Pictorial	Standard
Qwen2.5-VL-7b	27.62	31.93
Qwen2.5-VL-7b (+Ours)	**71.69**	**72.01**
Matcha	17.51	30.39
Matcha (+Ours)	33.90	35.87
ChartInstruct	21.18	29.22
ChartInstruct (+Ours)	56.09	58.54
ChartMLLM	19.90	26.08
ChartMLLM (+Ours)	61.26	61.36

Table 3: Performance comparison of models when interpreting charts with misleading labels.

5.6 Ablation Study

In ChartAlign, we use Dynamic Weight Averaging (DWA) to balance the alignment loss and consistency loss. Table 4 reveals how Dynamic Weight Averaging (DWA) impacts performance. DWA enables Qwen2.5-VL-7b to achieve superior results across metrics, producing the highest scores for both pictorial and standard charts and making it the state-of-the-art across all models. For ChartMLLM, DWA performs slightly lower than fixed weights (65.73% vs. 67.14% on pictorial charts and 66.85% vs. 67.56% on standard charts), demonstrating that manual parameter tuning only provides marginal benefits for this model. This minimal performance difference suggests DWA offers comparable results while eliminating the need for extensive hyperparameter optimization among different models.

5.7 Attention Visualization

To better understand how our ChartAlign method influences the attention patterns of MLLMs, we employed attention visualization techniques based on the VLM-Visualizer framework [46]. See Appendix 8.3 for the detailed visualization methodology.

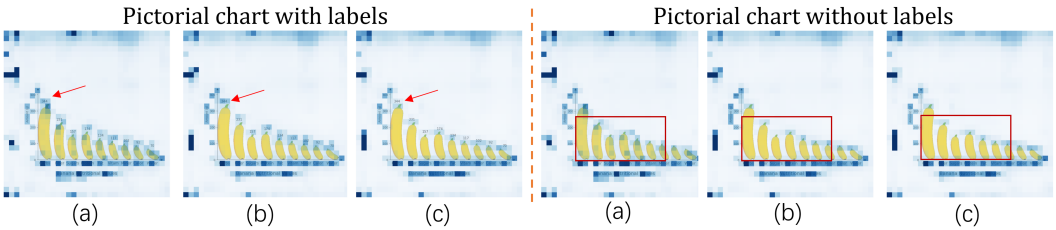

Figure 3: Visualization of MLLM's attention on charts. The left and right parts are pictorial charts with and without labels, respectively. (a)-(c) shows the attention of the original ChartMLLM, ChartMLLM+ChartAlign adapted to pictorial charts, and ChartMLLM+ChartAlign adapted to label-agnostic charts.

As shown in Figure 3, our ChartAlign method significantly alters attention patterns in beneficial ways. The base model (a) often focuses on decorative elements, while our ChartAlign-enhanced model shifts attention from decorative elements to critical data points and inflection points, enabling more accurate data extraction despite visually complex presentations. Similarly, the label-agnostic encoder (c) demonstrates a clear shift in attention away from textual data labels toward the visual elements and the axes of the chart. This confirms that our method successfully guide the model to rely on visual data representations rather than textual shortcuts, improving performance on unlabeled charts or those with misleading labels.

6 Limitations

While our ChartAlign framework demonstrates effectiveness, we acknowledge two areas for refinement. First, potential domain distribution differences exist between our synthetic dataset and real-world charts with their nuanced design elements, which may influence generalization to certain professional contexts. Second, our method can be further improved for better compatibility with feature-based token pruning architectures. While the current approach works well with encoders using consistent token representations or even using fixed token pruning (*e.g.*, Qwen2.5-VL), it fails to converge when applied to feature-based token merging methods like TinyChart[5]. This occurs because the extracted tokens from the source and target sets may differ substantially, causing misalignment in the feature matching process.

7 Conclusion

In this paper, we addressed a critical limitation of current MLLMs in chart understanding: the reliance on textual cues and standard visual elements in charts makes them fail to generalize well on semantically equivalent but visually diverse variants. To tackle this, we introduced **ChartPairs**, a dataset of paired charts for facilitating robust visual representation learning. Leveraging this dataset, we proposed **ChartAlign**, a novel instance-level alignment method for image encoders to ensure stronger visual consistency and a more faithful preservation of underlying data semantics. Our extensive evaluations demonstrate the effectiveness and plug-and-play nature of our method, effectively enhancing model capability in label-agnostic chart understanding and Pictorial chart understanding. In addition to chart understanding, our framework has the potential to be extended to other scenarios that require robust alignment of semantically equivalent instances.

References

[1] Ting-Yao Hsu, C Lee Giles, and Ting-Hao Huang. SciCap: Generating captions for scientific figures. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3258–3264, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[2] Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In Anna Rogers, Jordan Boyd-Graber, and

Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada, July 2023. Association for Computational Linguistics.

[3] Xingyu Lan, Yanqiu Wu, and Nan Cao. Affective visualization design: Leveraging the emotional impact of data. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):1–11, 2024.

[4] Xingchen Zeng, Haichuan Lin, Yilin Ye, and Wei Zeng. Advancing multimodal large language models in chart question answering with visualization-referenced instruction tuning. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):525–535, January 2025.

[5] Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. TinyChart: Efficient chart understanding with program-of-thoughts learning and visual token merging. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1898, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[6] Zhengzhuo Xu, Bowen Qu, Yiyan Qi, SiNan Du, Chengjin Xu, Chun Yuan, and Jian Guo. ChartMoE: Mixture of diversely aligned expert connector for chart understanding. In *The Thirteenth International Conference on Learning Representations*, 2025.

[7] Ahmed Masry, Mehrad Shahmohammadi, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. ChartInstruct: Instruction tuning for chart comprehension and reasoning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10387–10409, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[8] Jinyue Chen, Lingyu Kong, Haoran Wei, Chenglong Liu, Zheng Ge, Liang Zhao, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Onechart: Purify the chart structural extraction via one auxiliary token. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, page 147–155, New York, NY, USA, 2024. Association for Computing Machinery.

[9] Yifan Wu, Lutao Yan, Leixian Shen, Yunhai Wang, Nan Tang, and Yuyu Luo. ChartInsights: Evaluating multimodal large language models for low-level chart question answering. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12174–12200, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[10] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[11] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[12] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022.

[13] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, pages 8748–8763, 2021.

[15] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. Featured Certification.

[16] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[17] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc., 2022.

[18] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

[19] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[20] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26286–26296, 2024.

[21] Gen Luo, Yiyi Zhou, Yuxin Zhang, Xiawu Zheng, Xiaoshuai Sun, and Rongrong Ji. Feast your eyes: Mixture-of-resolution adaptation for multimodal large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[22] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR, 2023.

[23] Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Eisenschlos. MatCha: Enhancing visual language pretraining with math reasoning and chart derendering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12756–12770, Toronto, Canada, July 2023. Association for Computational Linguistics.

[24] Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. DePlot: One-shot visual language reasoning by plot-to-table translation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10381–10399, Toronto, Canada, July 2023. Association for Computational Linguistics.

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

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

[27] Victor Carbune, Hassan Mansoor, Fangyu Liu, Rahul Aralikatte, Gilles Baechler, Jindong Chen, and Abhanshu Sharma. Chart-based reasoning: Transferring capabilities from LLMs to VLMs. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 989–1004, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[28] Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. ChartAssistant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7775–7803, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[29] 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 Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1287–1310, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[30] Mubashara Akhtar, Nikesh Subedi, Vivek Gupta, Sahar Tahmasebi, Oana Cocarascu, and Elena Simperl. ChartCheck: Explainable fact-checking over real-world chart images. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13921–13937, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[31] Mubashara Akhtar, Oana Cocarascu, and Elena Simperl. Reading and reasoning over chart images for evidence-based automated fact-checking. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 399–414, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[32] Maxim Lisnic, Cole Polychronis, Alexander Lex, and Marina Kogan. Misleading beyond visual tricks: How people actually lie with charts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery.

[33] Leo Yu-Ho Lo, Ayush Gupta, Kento Shigyo, Aoyu Wu, Enrico Bertini, and Huamin Qu. Misinformed by visualization: What do we learn from misinformative visualizations? *Computer Graphics Forum*, 41(3):515–525, 2022.

[34] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomput.*, 312(C):135–153, October 2018.

[35] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance, 2014.

[36] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 97–105. JMLR.org, 2015.

[37] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):801–814, 2019.

[38] Xiangbo Shu, Guo-Jun Qi, Jinhui Tang, and Jingdong Wang. Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, page 35–44, New York, NY, USA, 2015. Association for Computing Machinery.

[39] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[40] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *The Eleventh International Conference on Learning Representations*, 2023.

[41] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[42] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1871–1880, 2019.

[43] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[44] Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[45] Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. Plotqa: Reasoning over scientific plots. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1516–1525, 2020.

[46] Steven Jia-Kai Zhang. Visualizing the attention of vision-language models. `https://github.com/zjysteven/VLM-Visualizer`, 2023. Accessed: 2024.

8 Appendix

8.1 Implementation Details

Parameter	ChartMLLM	ChartInstruct	Matcha	Qwen2.5-VL-7B	TinyChart
General Training Parameters					
Batch Size	16	16	16	16	16
Learning Rate	2e-5	2e-5	1e-5	2e-5	2e-5
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW
Weight Decay	0.01	0.01	0.01	0.01	0.01
Training Epochs	2	2	2	2	2
Warmup Steps	0	0	0	0	0
ChartAlign Specific Parameters					
λ_1, λ_2	1,1	DWA	1,1	DWA	DWA

Table 5: Training parameters for different models.

Table 5 presents training parameters for our four models. All models converged in 2 epochs with identical batch size (16) and optimizer settings, though Matcha uses a lower learning rate (1e-5) than

others (2e-5). ChartMLLM and Matcha employ fixed loss weights (1,1) while ChartInstruct and Qwen2.5-VL-7B use Dynamic Weight Averaging.

8.2 More results

8.2.1 Performance on Different Chart Types

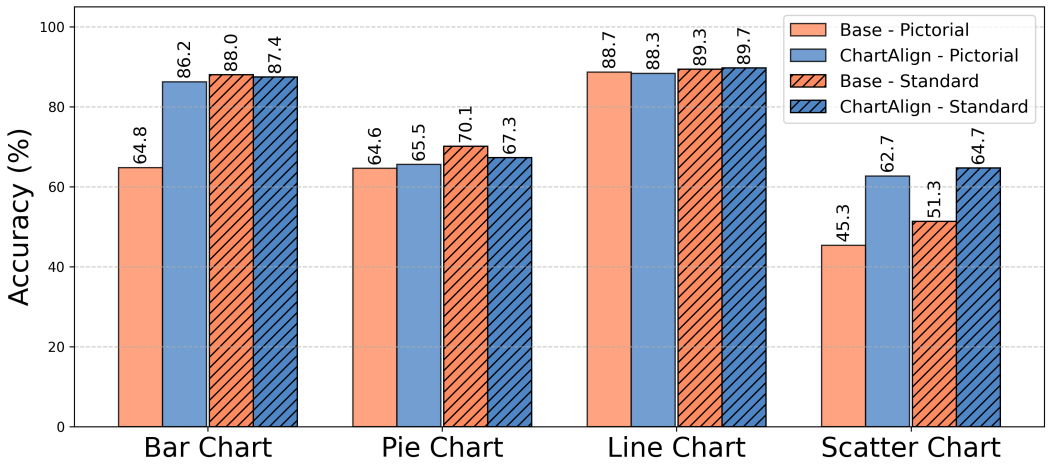

Figure 4: Performance comparison of Qwen2.5-VL-7B (Base) with and without ChartAlign across chart types on ChartPairs.

Figure 4 presents a comparative analysis of Qwen2.5-VL-7B's performance with and without ChartAlign across various chart types in ChartPairs. The results demonstrate significant improvement on pictorial bar charts, leveraging the original model's capabilities on standard bar charts. However, pie charts and line charts show only marginal improvements or slight decreases, attributed to the original model's already robust performance on both standard and pictorial variants of these chart types. Notably, scatter plots exhibit substantial performance enhancement, which is not inherent to this chart type itself, as the model outperforms the base version even on standard scatter plots. We attribute this improvement to the alignment of other chart types, particularly line charts, as they share similar construction principles.

8.2.2 Comparison of Different Training Methods

	Pictorial		Standard	
	w/ dl	w/o dl	w/ dl	w/o dl
Base	61.31	28.26	70.52	62.96
LLM	61.11	26.85	68.67	61.00
LLM+V	53.28	24.88	59.83	52.69
Ours	70.57	63.70	70.46	64.66

Table 6: Performance comparison (%) of training strategies using ChartMLLM as the base model. LLM-only tuning (LLM) uses LoRA to fine-tune only the language component, full model tuning (LLM+V) combines LoRA for LLM with full parameter tuning of vision encoder, while our ChartAlign approach targets only the vision encoder.

Tab. 6 compares different training approaches using ChartMLLM. Traditional supervised fine-tuning (LLM+V) performs substantially worse than the base model (pictorial charts: 61.31% $\rightarrow$ 53.28%, -13.1%$\downarrow$; standard charts: 70.52% $\rightarrow$ 59.83%, -15.2%$\downarrow$). This decline stems from limited training data causing overfitting. In contrast, our ChartAlign approach, targeting only the vision encoder through knowledge distillation, preserves language capabilities while significantly enhancing visual feature extraction (pictorial charts: 61.31% $\rightarrow$ 70.57%, +15.1%$\uparrow$).

8.3 Visualization Methodology

To provide deeper insights into how our ChartAlign method affects the model's attention distribution when processing chart images, we developed a comprehensive visualization pipeline based on the VLM-Visualizer framework [46]. Our approach extracts and visualizes cross-modal attention patterns to reveal what visual elements the model focuses on when generating text about chart data.

8.3.1 Visualization Process

The visualization process works in three key stages:

Stage 1: Vision Encoder Attention Extraction. We extract the attention maps from all transformer layers in the CLIP vision encoder and aggregate them across layers. Specifically, for each transformer layer l, we compute the attention matrix $A^l \in \mathbb{R}^{N \times N}$, where N is the number of image tokens. These attention matrices are then aggregated across all L layers:

$$A_{\text{agg}} = \sum_{l=1}^{L} A^l \tag{6}$$

This produces a two-dimensional attention score for each image token that can be directly mapped to spatial locations in the original image through bilinear interpolation.

Stage 2: Cross-Modal Attention Integration. During text generation in the LLM, each newly generated token t_i has attention scores $\alpha_i \in \mathbb{R}^M$ toward all previous tokens (including image tokens), where M is the total number of tokens in the input sequence. We extract these one-dimensional scores and use them as weights to compute a weighted sum of the image token attention maps:

$$H_i = \sum_{j=1}^{N_{\text{img}}} \alpha_{i,j} \cdot A_{\text{agg},j} \tag{7}$$

where N_{img} is the number of image tokens, and $A_{\text{agg},j}$ is the attention map for the j-th image token. This produces the visualization heatmap H_i for the i-th generated token.

Stage 3: Response-level Aggregation. To obtain the overall attention distribution for the entire response, we average the heatmaps across all generated tokens:

$$H_{\text{final}} = \frac{1}{T} \sum_{i=1}^{T} H_i \tag{8}$$

where T is the total number of tokens in the model's response. This final aggregated heatmap H_{final} reveals the model's overall attention distribution when answering chart-related queries, providing insights into which visual elements are most influential for the model's understanding and reasoning process.

Score Normalization. To mitigate the influence of extreme outliers in the attention distribution, we apply a smoothing operation prior to visualization. Specifically, we truncate the upper tail of the score distribution by replacing the top five maximum values with the fifth-highest value (Winsorization). This preprocessing step ensures more stable color mapping while preserving the relative attention patterns across the image. The normalized scores are then projected to a continuous color space for visualization.

9 Potential Social Impact

9.1 Positive Societal Impacts

Enhanced Data Accessibility: By improving MLLMs' ability to interpret visually diverse charts, our work can help individuals with visual impairments better understand and utilize chart data, promoting equality in information access.

Combating Misleading Information: By combining our ChartAlign for text-free analysis with standard models, we can identify misleading data labels, helping detect and reduce manipulation in data visualizations and combating the spread of misinformation.

9.2 Negative Societal Impacts

Technology Dependence: Overreliance on automated chart interpretation may weaken human abilities to analyze and think critically about data visualizations.

Privacy and Security Risks: Enhanced chart understanding capabilities could be used to extract information from sensitive documents containing charts, increasing the risk of data breaches.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes] .

   Justification: We have made the main claims in the abstract and introduction to accurately reflect the paper's contributions and scope.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In section 6, we discuss the limitations of the work performed by the authors.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: We have provided the full set of assumptions and a complete (and correct) proof for our theoretical result.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided all the information needed to reproduce the main experimental results of the paper, and we will release the code and data.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide open access to the data and code, and sufficient instructions to faithfully reproduce the main experimental results.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have specified all the training and test details necessary to understand the results, and we have provided the code and data.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars in the paper because it is too computationally expensive for MLLMs. Also, other outstanding paper related to chart understanding are not reporting error bar, *e.g.*ChartMoE [6] on ICLR 2025.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We have provided sufficient information on the computer resources needed to reproduce the experiments.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have followed the NeurIPS Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed both potential positive societal impacts and negative societal impacts of the work performed in appendix.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: We do not think that the data or model in our paper poses a high risk for misuse.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly credited the creators of the assets and included the license and terms of use.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have provided documentation for the new assets.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not involve crowdsourcing or research with human subjects in this paper.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve research with human subjects in this paper.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We do not use LLMs as an important, original, or non-standard component of the core methods in this research.