ChartAB: A Benchmark for Chart Grounding & Dense Alignment

Aniruddh Bansal, Davit Soselia, Dang Nguyen, Tianyi Zhou
University of Maryland, College Park
{ani01, dsoselia, dangmn}@umd.edu
Project: https://github.com/tianyi-lab/ChartAlignBench

Abstract

Charts play an important role in visualization, reasoning, data analysis, and the exchange of ideas among humans. However, existing vision-language models (VLMs) still lack accurate perception of details and struggle to extract fine-grained structures from charts. Such limitations in chart grounding also hinder their ability to compare multiple charts and reason over them. In this paper, we introduce a novel "**ChartA**lign **B**enchmark (ChartAB)" to provide a comprehensive evaluation of VLMs in chart grounding tasks, i.e., extracting tabular data, localizing visualization elements, and recognizing various attributes from charts of diverse types and complexities. We design a JSON template to facilitate the calculation of evaluation metrics specifically tailored for each grounding task. By incorporating a novel two-stage inference workflow, the benchmark can further evaluate VLMs' capability to align and compare elements/attributes across two charts. Our analysis of evaluations on several recent VLMs reveals new insights into their perception biases, weaknesses, robustness, and hallucinations in chart understanding. These findings highlight the fine-grained discrepancies among VLMs in chart understanding tasks and point to specific skills that need to be strengthened in current models.

1 Introduction

Recent large multimodal models (LMMs), such as vision-language models (VLMs), have achieved remarkable breakthroughs in aligning the visual modality with language models, enabling challenging language-level reasoning on visual input signals and opening the door to a wide range of applications that naturally rely on interactions between the two modalities (Alayrac et al., 2022; Li et al., 2023; Liu et al., 2023b). One critical class of applications is chart understanding and reasoning, which has broad use in finance, data science, mass media, biology, and other scientific domains where ideas and information are communicated through visualizations. In these applications, measuring numerical values in charts, comparing visual elements (e.g., bars or curves), mapping correspondences between colors, numbers, names, or markers, and recognizing attributes are essential skills for downstream tasks. Most of these tasks require accurate grounding of the structured details in charts. Moreover, dense alignment of elements across multiple charts is also a widely needed skill in practical scenarios. These challenges present new open problems for VLMs.

Instead of focusing on charts, existing VLMs have primarily been pretrained and finetuned on natural images and common questions/instructions, which are not fully compatible with chart understanding tasks (Yao et al., 2024; Laurençon et al., 2024). Unlike perceiving objects' shapes, poses, and semantic meanings in natural images, accurate measurement and comparison of geometric/graphic components, understanding of their structure and layout, and manipulation of their positions and rich textual content are more critical for perception and reasoning with chart images. However, it remains challenging for VLMs to acquire these capabilities, often leading to hallucinations and misinterpretations in chart-centric tasks (Masry et al., 2022; Xia et al., 2024).

Despite the recent growing interest in chart-related tasks, existing VLMs and benchmarks specifically designed for charts usually focus on simple QA tasks (Masry et al., 2022; 2025; Wang et al., 2024b; Li & Tajbakhsh, 2023), which cannot comprehensively assess the capabilities of VLMs in grounding and understanding chart components for more general-purpose tasks. Moreover, the alignment of

layouts and components across multiple charts has not been explored in previous work. Hence, there remains a lack of benchmarks dedicated to evaluating these critical skills.

In this paper, we take the first step toward systematically evaluating and analyzing general-purpose VLMs on chart grounding and multi-chart dense alignment. We formally categorize the information to be grounded in a chart into two dimensions: (1) **data**, and (2) **attributes** (e.g., colors, styles, legends, sizes, positions) that define the visualization design, components, and layout. We define the *chart grounding task* as extracting both the underlying data table and the associated attributes from a chart image, and the *dense alignment task* as identifying correspondences and differences between two charts. Together, these tasks represent fundamental capabilities and critical subroutines required for a wide range of chart-centric applications.

To this end, we develop a comprehensive benchmark using pairs of similar charts to evaluate model performance on the two tasks with respect to each type of information in the two categories. To create a pair of similar charts, we perturb an existing chart by randomly modifying (1) one or a few data cells in the data table and/or (2) an attribute in the script used to generate the original chart. To maximize the potential of VLMs and evaluate their full capabilities, we propose a multi-stage information extraction and query pipeline. In this pipeline, VLMs are first queried with a grounding task targeting specified information in each chart, followed by a comparison of the grounding results between the two charts. The pipeline leverages structured JSON templates to guide the grounding and alignment of different types of information. In addition, we introduce several novel evaluation metrics that account for the symmetry and ambiguity inherent in various types of information, thereby enabling more reliable quantitative comparisons across different VLMs.

Our analysis reveals the weaknesses of existing VLMs in chart perception and understanding for dense grounding and alignment. The observed errors highlight their biases and hallucinations regarding certain chart components, offering critical insights for improving VLMs. The evaluation results further show how differences across models, chart types, and queried data/attributes influence benchmarking performance. In addition, we assess the robustness of VLMs in data grounding and alignment under different attribute variations, such as changes in chart type or color schemes.

Our contributions and novelties are summarized as follows:

- We introduce the first comprehensive benchmark, "ChartAB" to systematically evaluate VLMs' capabilities in dense grounding and alignment of data and attributes in multiple chart images.
- We propose a holistic evaluation suite, including a multi-stage pipeline converting charts into JSON files with specific templates for data/attributes grounding, and a rating scheme of the grounding/alignment performance based on VLMs' answers.
- Our evaluation and analysis of existing VLMs reveal their weaknesses in fine-grained chart understanding, highlight hallucinations, and expose biases in their vision encoders when perceiving critical chart features and structures.
- We evaluate VLMs' robustness on data grounding and alignment under perturbations of attributes. It provides novel insights for the design of high-quality charts.

2   RELATED WORK

VLMs for Charts.   Vision-language models have shown significant advancements in chart understanding tasks. They can be broadly classified into (1) general-purpose multimodal models and (2) chart-specialized models. General-purpose models include proprietary ones (Hurst et al., 2024) and open-source ones (Abdin et al., 2024; Chen et al., 2024; Liu et al., 2023a; Bai et al., 2025). Chart-specialized models (Zhang et al., 2024b; Masry et al., 2024; Xia et al., 2024; Meng et al., 2024) demonstrate strong performance on chart benchmarks; however, they are limited by instruction tuning on specific tasks, which restricts dense-level understanding, and are further hindered by incompatible pipelines that often rely on predefined routines to handle task requirements.

Chart Understanding Benchmarks.   Current chart benchmarks evaluate VLMs on specific tasks including question answering (Methani et al., 2020; Masry et al., 2022), summarization (Kantharaj et al., 2022b), explanation-generation (Kantharaj et al., 2022a). Multi-task benchmarks including ChartLlama Han et al. (2023), ChartX Xia et al. (2024) perform agglomeration of various modalities

(like chart data, description, summary) for the downstream tasks. Recent works specifically focus on expanding QA scope to overcome increased saturation by VLMs, for example CharXiv Wang et al. (2024b) focuses on charts in research papers, SciGraphQA Li & Tajbakhsh (2023) evaluates multi-turn QA, MultiChartQA Zhu et al. (2024) evaluates multi-hop reasoning on multiple related charts, ChartQAPro Masry et al. (2025) includes diverse visualizations such as dashboards, infographs, and flexible questions (hypothetical, unanswerable).

Visual Grounding. The dense-level understanding abilities of VLMs have been extensively enhanced through visual grounding. DePlot Liu et al. (2022) trained a transformer for image-to-CSV generation, introducing a novel table comparison method for evaluation. StructChart Xia et al. (2023) proposed module-based augmentation for efficient grounding of chart data and plot code in downstream applications. Beyond charts, the Grounded-SAM model (Ren et al., 2024) leverages Grounding-DINO (Liu et al., 2024) for improved dense-level open-set object tracking. BLIP-2 Li et al. (2023) has been widely integrated into VLMs for VQA-related tasks. LLaVA-Grounded Zhang et al. (2024a) enables detailed text descriptions of multi-object natural images by leveraging image–text grounding for instruction tuning.

Multi-Image Reasoning. Multiple benchmarks have been developed to evaluate VLMs on multi-image reasoning. MMMU Yue et al. (2024) includes interleaved examples with multiple images from medical, cartoon, art, and technical domains. MUIRBench Wang et al. (2024a) focuses on multi-chart diagram QA but is limited to coarse-level understanding. MMIR Zhao et al. (2024) addresses chart understanding through cross-modal alignment, i.e., plotting-code correctness relative to the chart image. MileBench Song et al. (2024) introduces semantic understanding tasks involving text-rich images, emphasizing text extraction and comprehension in OCR, documents, and slides.
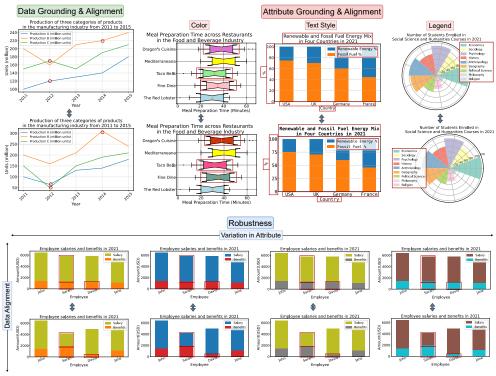
3 ChartAB: CHART GROUNDING AND ALIGNMENT BENCHMARK

Figure 1: **Examples of paired charts for `ChartAB` tasks.** `ChartAB` evaluates dense grounding and alignment capabilities of VLMs on chart images. (1) Paired charts in each *Data Grounding & Alignment* task differ in a few visualized data values. (2) Paired charts in each *Attribute Grounding & Alignment* task differ in a visualization attribute, e.g., color, legend position, or text style. (3) Each *Robustness* task contains multiple variants of the same chart-pair for *Data Alignment*, with different attributes (e.g., colors) across the variants.

We introduce `ChartAB`, the first benchmark designed to evaluate vision-language models (VLMs) on dense level chart understanding. The benchmark focuses on three core capabilities essential to chart reasoning: (1) *grounding*: extracting structured information from a single chart image, (2) *alignment*: identifying fine-grained differences between a pair of similar charts, and (3) *robustness*: assessing the stability of alignment performance under variations in chart appearance. These capabilities serve as cornerstones for a wide range of downstream applications. We develop a novel two-stage pipeline that can isolate and rigorously evaluate them. Thereby, `ChartAB` offers a deeper diagnostic suite of VLMs' perceptual accuracy, reasoning limits, and alignment behavior in structured visual domains.

3.1 DATASET TAXONOMY AND CONSTRUCTION

We construct `ChartAB` from ChartX Xia et al. (2024) as the source dataset. It encompasses diverse chart types from various domains, including commerce, industry, lifestyle, society, and culture, and provides both CSV data and plotting code for each chart. We list the taxonomy of `ChartAB` in Table 1. For each chart, we extract dense annotations of two types of fine-grained information: (1) *Data*: The underlying data table that the chart visualizes. (2) *Attributes*: The visual attributes that defines the appearance of the

Table 1: **Task Taxonomy in `ChartAB`**, which is composed of three types of tasks defined on different data cells and attributes.

Task Type	Data			Attributes				
	1-Cell	2-Cell	3-Cell	Color	Legend	Text Style		
						Size	Weight	Font Family
Grounding		•		•	•	•	•	•
Alignment	•	•	•	•	•		•	
Robustness	•	•	•					

chart, e.g., *color*, *legend*, and *text Style*. In particular, *color* refers to the colors of the visual elements as bars, lines, or boxes in charts. *Legend* refers to the position of the chart legend. *Text Style* captures the textual characteristics in four chart regions: title, legend, axis labels, and axis ticks. These characteristics include textual size, weight (lightness/boldness), and font family (e.g., Times New Roman).

Section 3.2 introduces three types of tasks built upon the dense annotations. Grounding tasks aim to extract these dense labels, while robustness tasks evaluate grounding performance under perturbations of attributes. Alignment tasks introduced aim to identify the differences between two similar charts. To create pairs of similar charts, we draw an image from the ChartX, apply controlled modifications in the plotting code, and execute the code to render an variant of the original chart. Each chart's source data (CSV file) and plotting script are provided in ChartX, ensuring precise ground-truths.

Figure 2: **Statistics of `ChartAB`.** `ChartAB` includes 9,000+ instances curated for tasks below: (1) Paired charts for *Data Grounding & Alignment* differ in one to three data cells; (2) Paired charts for *Attribute Grounding & Alignment* differ in color, legend position, or text style; (3) *Robustness* task includes multiple pairs that share identical differences in data but differ in certain attributes.

Figure 1 provides several examples of different tasks, while Figure 2 reports the statistics of these tasks. `ChartAB` covers nine diverse chart types with different data and attribute perturbations: (1) *simple charts*: bar chart, bar-numbered chart, line chart, and line-numbered chart; (2) *complex charts*: 3D chart, box chart, radar chart, rose chart, and multi-axes chart. More details about chart data curation are provided in A.3.
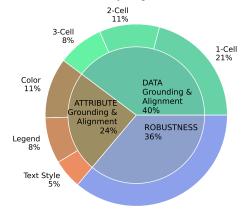
3.2 EVALUATION TASKS

Grounding of Single Charts Dense grounding of chart elements requires the extraction of precise semantic information from chart images. However, general-purpose VLMs are trained to mainly focus on global visual features or major objects in scenes. When applied to charts, they often fall short of perceiving the details (Xu et al., 2023), which are crucial for chart reasoning. Prior works primarily evaluate VLMs' chart understanding capabilities via QA tasks, which do not fully capture their semantic grounding or reflect their cross-modal inconsistencies (Huang et al., 2024). To ensure

interpretable and compositional reasoning, we need to examine whether VLMs can ground the chart information in textual form.

We formalize *Grounding* as the conversion of a chart image into a structured textual representation of data or attributes. As shown in Table 1, we assess this capability through the following tasks: (1) Data Grounding, (2) Color Grounding, (3) Legend Grounding, (4) Text Style Grounding (subtasks: Size, Weight, Font Family). *Data Grounding* requires the VLM to generate a standard CSV representation of the data table. We provide a JSON template for tasks requiring Attribute Grounding (*Color/Legend/Text Style*) and prompt the model to generate a JSON representation.

Grounding the chart image into textual form isolates the model's perceptual ability from downstream prompt variation or instruction complexity. This helps build a foundation for the subsequent dense alignment and QA tasks, while also enabling failure analysis of VLM in perceiving chart components.

Dense Alignment between Two Charts While single chart grounding evaluates a model's perception of details in a given chart, multi-chart reasoning in practice often requires comparing similar charts to detect and analyze the differences among them. To evaluate this capability, we define a dense alignment task where the model identifies fine-grained discrepancies between two charts. Crucially, this task builds on grounded representations, allowing us to isolate and evaluate comparative reasoning for given chart pairs. As shown in our ablation studies (A.6.2), direct alignment without grounding yields significantly weaker performance, highlighting the necessity of grounding for subsequent dense alignment.

We formalize *Dense Alignment* as a comparison of two chart images that differ in local details of data or attributes. As shown in Table 1, we assess this capability via the following tasks: (1) *Data Alignment*, (2) *Color Alignment*, (3) *Legend Alignment*, (4) *Text Style Alignment*. *Data Alignment* task is further divided into subtasks: *1-cell*, *2-cell*, and *3-cell*, which perform dense alignment of data for chart images that differ in 1, 2, and 3 data points, respectively. Each alignment task challenges the model to identify the set of divergent content and produce a structured JSON listing these differences.

Robustness of Data Alignment to Attribute Variation Using VLMs for real-world understanding of charts requires analyzing charts in diverse visual forms, i.e., diverse attributes (color/text style/legends) presence for similar types of data, often due to differing plotting tools. Moreover, past work shows the sensitivity of VLM's chart understanding under attribute changes (Guo et al., 2024). Hence, it motivates the evaluation of VLM's chart understanding consistency across noise, style shifts, and design variations due to variations in attributes.

We thus formalize *Robustness* of Data Alignment to variation in Attributes (Color/Legend/Text Style). To perform the task, each instance contains five pairs of chart variants created from the same pair of charts. Each pair visualizes the same source data and maintains identical data differences as the other four pairs, but their attributes (e.g., color of bars) vary across the five pairs.

Effects of Dense Grounding & Alignment on Downstream QA Tasks Practical applications of VLMs on chart-related tasks often require complex reasoning, in which dense grounding & alignment usually serve as foundational building blocks and the cornerstone of various downstream tasks. On the other hand, grounding/alignment errors are common reasons for many reasoning failures of VLMs on charts. To demonstrate the importance of dense grounding/alignment skills, we evaluate VLMs on QA tasks, the most widely applied category of downstream tasks, and investigate the correlation between QA performance and the grounding/alignment quality scores. To this end, our study is conducted on QA tasks from ChartX (Xia et al., 2024) that have single-word answers derived from the grounded CSV tables.

3.3 A Two-Stage Evaluation Pipeline

We propose a two-stage evaluation pipeline inspired by the multi-step approach of SOTA reasoning models, for example, color alignment by o4-mini OpenAI (2025) in Figure 3. The model's reasoning takes two steps: grounding the box colors in each chart, followed by dense alignment (comparison) of their grounded colors. This two-stage strategy performs complex, finer-level reasoning by ground-then-compare subtasks with efficient element-wise comparisons. It thus mitigates hallucinations and outperforms the one-stage strategy of GPT-4o, validating the importance of dense grounding for other tasks.

In our evaluation pipeline, the prompt in each stage consists of natural language instructions with a task-specific JSON template defining the output format. This enables better inswtruction following and flexible output parsing and evaluation. As shown in Figure 4, The *first-stage* performs grounding of data or certain attributes in the given charts. Such well-formatted element-wise representation facilitates subsequent dense alignment and QA tasks. The *second-stage* compares the grounding results of the two charts from the first stage and produces a JSON file to list the dense alignment results.
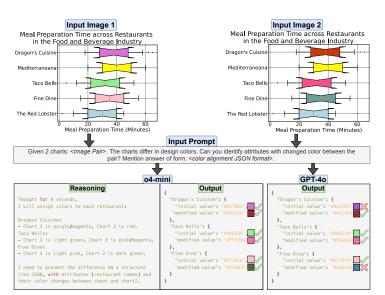

Figure 3: **Two-stage color alignment by o4-mini.** The o4-mini model automatically decomposes the task into a grounding step for the colors in each chart, followed by an output prediction of the alignment. This two-stage reasoning yields a more accurate result than GPT-4o, which performs alignment directly without intermediate grounding.

The second stage is critical to evaluating end-to-end alignment as it requires VLMs to perform semantic comparison over grounded outputs, beyond surface-level extraction. Compared to one-stage approaches, it mitigates grounding ambiguities and collects additional context, offering a more human-like assessment of alignment ability. More details of the pipeline are discussed in A.4.

3.4 EVALUATION METRICS

Dense Grounding performance is evaluated by the precision of the detected semantic elements in a given chart, e.g., values of visualized data, color of bars, legend position, font size. In the experiments, we report (1) *Legend position* grounding's confusion matrix in Figure 8; (2) Text-style grounding accuracy in Figure 6; (3) *Color* grounding's L2 error of RGB values in Figure 7; and (4) *Data* grounding performance in Figure 9b is evaluated by the precision of predicted CSV using the SCRM metric introduced in StructChart (Xia et al., 2023).

Dense Alignment performance is evaluated across four task categories: *data alignment* (subtasks: 1-cell/2-cell/3-cell), *color alignment*, *text style alignment*, and *legend alignment*. For each chart pair, the model is prompted to output a JSON file that lists the differences on possible attributes and their own values. The performance on the first three tasks is evaluated by a key-value alignment score, which assess the capability to identify the different elements (keys) between two charts and their associated values. In contrast, legend alignment score mainly focuses on comparing the different spatial positions of legends in two charts (values only) because the key (i.e., the position) is unique and fixed. More details of the keys and values are provided in Table 2, while the concrete definitions of the metrics are introduced in A.5.1.

Robustness of data alignment to the variations of different visualization attributes, e.g., colors, legend positions, text style, is evaluated by the standard deviation of data alignment scores over multiple variations of the original chart pairs. We evaluate the robustness score under the variation of each attribute, and report the averaged scores over chart pairs. More details of the robustness score are provided in A.5.2.

Grounding/Alignment affects QA Performance To further analyze the impact of grounding/alignment quality on downstream QA tasks, we evaluate QA accuracy by following the protocols in ChartX (Xia et al., 2024): string-based answers require an exact match, while numerical values are considered correct if they fall within a 5% error margin; and investigate its correlation with the grounding/alignment performance. To this end, we adopt a two-stage QA that firstly extracts a CSV (table) file from a chart (data grounding), and then answers the question given the grounding result.
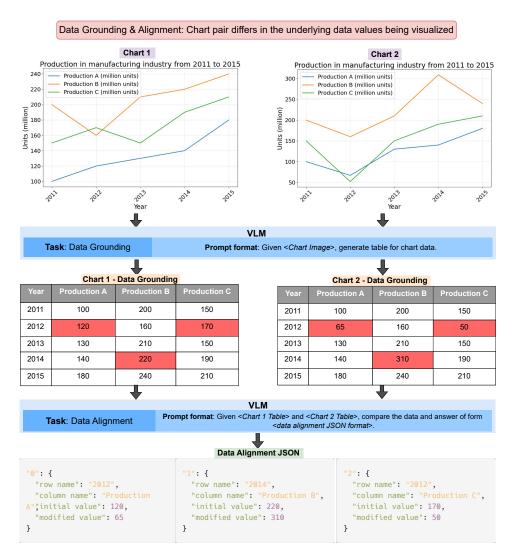
Figure 4: **Two-Stage Evaluation Pipeline for *Data Grounding & Alignment* in `ChartAB`.** The first stage focuses on grounding the data visualized by each chart in a CSV table, while the second stage focuses on alignment, which aims to allocate the difference between the two tables and output a JSON file listing the different cells. The other two categories of tasks in `ChartAB` also adopt similar multi-stage pipelines, detailed in Figures 15, 16, 17 of the Appendix.

We analyze how this two-stage QA's accuracy and its difference to the ordinary one-stage QA's accuracy vary with grounding/alignment quality, which results are reported in Figure 9.

4   EXPERIMENTS & ANALYSIS

We evaluated GPT-4o (Hurst et al., 2024) and four open-source VLM families: Phi-3.5 vision-instruct (Abdin et al., 2024), InternVL-2.5 (Chen et al., 2024), LLaVA-1.6 (Liu et al., 2023a), QWEN-2.5 VL (Bai et al., 2025). We also evaluated chart-specialized VLMs, including TinyChart (Zhang et al., 2024b) and ChartGemmap Masry et al. (2024). However, as discussed in Section 2, their task-specific training leads to a collapse of general instruction following capabilities and fails to output the JSON format required by evaluation. Further discussion and ablation study are provided in A.6.1 and A.6.2.

Finding 1

VLMs' dense grounding and alignment of data/color are not satisfying on complex charts.

Compared to simpler and more common charts, e.g., bar/line charts and numbered bar/line charts, dense grounding/alignment on complex charts such as 3D/box/radar/rose/multi-axes charts with more components and irregular layouts is more challenging to most VLMs. Despite the similar alignment performance for *legend* (Figure 12a) and *text-style* (Figure 12b) between simple vs. complex charts, the *color* and *data* alignment (Figure 5) on complex charts are much poorer than those on simple charts. The color grounding requires identifying each constituent's visual encoding and corresponding color, while the data grounding needs to find the mapping from visual encoding to numeric values. Hence, complex layouts with more components make these tasks more difficult. In contrast, identifying the position of legends and text styles (which both have limited options) is easier and less affected by the chart complexity.
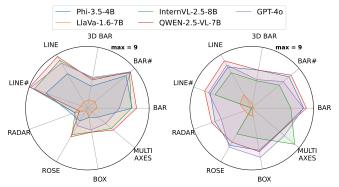

Figure 5: **Left:** Comparing VLMs on *Data Alignment* tasks on paried charts with **one-cell** difference. Llava-1.6 performs worse than most other VLMs, while QWEN-2.5-VL outperforms GPT-4o on most chart types. **Right:** *Color alignment* on fine-grained visual elements (e.g., bars, lines, sectors) between two charts. Most VLMs perform better on simpler and more common charts, e.g., line/bar charts. Related discussion beneath Finding 1.

> **Finding 2**
>
> VLMs' text-style grounding and alignment performance is poor in general, and it varies across text size, weight, and font family.

Figure 6 shows that most VLMs fail to detect the correct text size and font family, suffering from a $<20\%$ accuracy (except GPT-4o's performance on font family grounding). These indicate a lack of knowledge on these two text attributes. VLMs' performance on text weight ((light/normal/bold)) is much better ($\sim 60\%$) and close to each other, but still not satisfying. Although LLMs can select reasonable text sizes in code generation for plots, they tend to rely on the default sizes in their priors or relative sizes to other chart components. They still lack sufficient capability to identify exact text sizes in chart images.
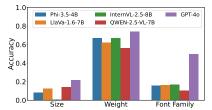

Figure 6: *Text-style grounding* on size, weight, and font family. The low accuracy of most VLMs highlights the lack of style knowledge (Finding 4).

> **Finding 3**
>
> VLMs' weak color recognition ability.

As shown in Figure 7, all models' color grounding error (L2 distance in RGB space) has a median exceeding 50. This implies their inability to understand color shades beyond common ones, e.g., red, blue, green, etc., which exposes their weaknesses in color recognition.

The lack of color understanding affects the perception of detailed differences in charts and leads to misalignment in color-conditioned reasoning tasks. Consequently, the VLMs' performance in color alignment tasks (Figure 5) is consistent with that on color grounding. These results suggest to improve the color understanding capability by adding more color-sensitive data to VLM training.
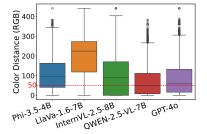

Figure 7: *Color grounding*'s L2 error in the RGB space, which median over VLMs >50 implies their weaknesses in color recognition (Finding 3).
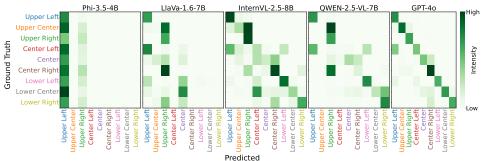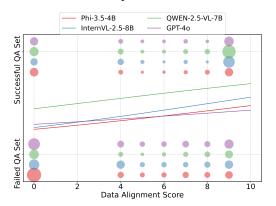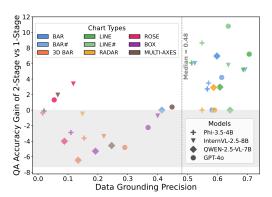
Figure 8: Confusion matrix of *legend position grounding*. The dark non-diagonal entries show the fail patterns and imply the biases of incorrectly identifying position-i as position-j. Phi-3.5 exhibits a severe bias towards the *upper-left* position while GPT-4o shows the minimal bias. More discussion is provided below Finding 2.

Finding 4

Spatial reasoning bias: Most VLMs suffer from biases when allocating the position of legends.

The grounding of the legend's position (Figure 8) suffers from a strong bias of pretrained VLMs. The Phi-3.5 model shows the strongest prior towards the *upper-left* position. The 7-8B scale VLMs, e.g., LlaVa-1.6, Inten-VL-2.5, QWEN-2.5-VL, all show a similar level of bias but towards the *upper-right* position instead. The GPT-4o model exhibits the minimal bias among all evaluated VLMs. The grounding bias strongly affects the legend alignment (Figure 12a) where Phi-3.5 performs the worst, GPT-4o has the best performance, while the other 3 models' performance is between them.

(a) Data Alignment correlates with QA performance.

(b) Data Grounding's impact on QA Performance.

Figure 9: **(a)** shows that the failed (successful) QA tasks decrease (increase) with the data alignment score, underscoring the importance of data alignment capability of VLMs on downstream chart reasoning tasks. **(b)** shows that precise (poor) data grounding leads to positive (negative) gain on QA tasks, indicating the importance of data grounding on downstram tasks. More discussion can be found beneath Finding 6.

Finding 5

Poor (precise) grounding and alignment degrade (improve) downstream QA performance.

Figure 9b demonstrates that precise (poor) grounding of chart-visualized data boosts (degrades) QA performance. It validates grounding as a gateway to extract structured data from charts for reliable downstream reasoning. Notably, the greatest gains are achieved on simple chart types (bar/line charts and numbered bar/line charts) due to better numeric understanding of these charts' visualized data, as discussed in Finding 1. Figure 9a shows a steady rise of QA accuracy (predicted) with the data alignment score, demonstrating the importance of dense chart understanding to QA reasoning. These findings position grounding and alignment as essential prerequisites for chart reasoning.

> **Finding 6**
>
> VLMs follow the scaling law on most dense alignment tasks.

As shown in Figure 10, we observed a consistent scaling law across most dense alignment subtasks, except for Text-Style Alignment. The deviation arises from the relatively greater complexity of the JSON template in this task, which led to a significantly higher number of failures where InternVL-2.5 produced incorrect JSON formats.

5 CONCLUSION

We introduce `ChartAB`, the first benchmark for fine-grained chart grounding and multi-chart dense alignment in vision–language models (VLMs). Our evaluations across diverse chart types reveal persistent challenges, including perceptual bias, weak attribute understanding, and limited spatial reasoning especially on complex visual representations. Experiments with our
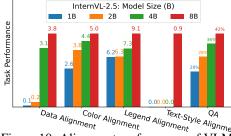

Figure 10: Alignment performance of VLMs with different sizes from the InternVL-2.5 family. Results of other VLMs are reported in Appendix 11.

novel two-stage pipeline show effectiveness of intermediate grounding in improving dense alignment, and the impact of grounding and alignment accuracy for enhance downstream question answering, establishing these capabilities as essential foundations for robust chart understanding.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

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, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/960a172bc7fbf0177ccccbb411a7d800-Abstract-Conference.html.

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

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

William W Cohen, Pradeep Ravikumar, Stephen E Fienberg, et al. A comparison of string distance metrics for name-matching tasks. In *IIWeb*, volume 3, pp. 73–78, 2003.

Grace Guo, Jenna Jiayi Kang, Raj Sanjay Shah, Hanspeter Pfister, and Sashank Varma. Understanding graphical perception in data visualization through zero-shot prompting of vision-language models. *arXiv preprint arXiv:2411.00257*, 2024.

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.

Wen Huang, Hongbin Liu, Minxin Guo, and Neil Zhenqiang Gong. Visual hallucinations of multimodal large language models. *arXiv preprint arXiv:2402.14683*, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Shankar Kantharaj, Xuan Long Do, Rixie Tiffany Ko Leong, Jia Qing Tan, Enamul Hoque, and Shafiq Joty. Opencqa: Open-ended question answering with charts. *arXiv preprint arXiv:2210.06628*, 2022a.

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

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models?, 2024. URL https://arxiv.org/abs/2405.02246.

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 *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Shengzhi Li and Nima Tajbakhsh. Scigraphqa: A large-scale synthetic multi-turn question-answering dataset for scientific graphs. *arXiv preprint arXiv:2308.03349*, 2023.

Fangyu Liu, Julian Martin 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. *arXiv preprint arXiv:2212.10505*, 2022.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023b. URL http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pp. 38–55. Springer, 2024.

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

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

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

Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. Chartassisstant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. *arXiv preprint arXiv:2401.02384*, 2024.

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

OpenAI. Openai o3 and o4-mini system card. Technical report, April 2025. System card covering multimodal and reasoning capabilities, safety evaluation, tool use, and performance benchmarks.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

Dingjie Song, Shunian Chen, Guiming Hardy Chen, Fei Yu, Xiang Wan, and Benyou Wang. Milebench: Benchmarking mllms in long context. *arXiv preprint arXiv:2404.18532*, 2024.

Fei Wang, Xingyu Fu, James Y Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, et al. Muirbench: A comprehensive benchmark for robust multi-image understanding. *arXiv preprint arXiv:2406.09411*, 2024a.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, et al. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697, 2024b.

Renqiu Xia, Bo Zhang, Haoyang Peng, Hancheng Ye, Xiangchao Yan, Peng Ye, Botian Shi, Yu Qiao, and Junchi Yan. Structchart: Perception, structuring, reasoning for visual chart understanding. *arXiv preprint arXiv:2309.11268*, 2023.

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.

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

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm-v: A gpt-4v level mllm on your phone, 2024. URL `https://arxiv.org/abs/2408.01800`.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024.

Hao Zhang, Hongyang Li, Feng Li, Tianhe Ren, Xueyan Zou, Shilong Liu, Shijia Huang, Jianfeng Gao, Leizhang, Chunyuan Li, et al. Llava-grounding: Grounded visual chat with large multimodal models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024a.

Li Zhang, Shuo Zhang, and Krisztian Balog. Table2vec: Neural word and entity embeddings for table population and retrieval. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pp. 1029–1032, 2019.

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

Bingchen Zhao, Yongshuo Zong, Letian Zhang, and Timothy Hospedales. Benchmarking multi-image understanding in vision and language models: Perception, knowledge, reasoning, and multi-hop reasoning. *arXiv preprint arXiv:2406.12742*, 2024.

Zifeng Zhu, Mengzhao Jia, Zhihan Zhang, Lang Li, and Meng Jiang. Multichartqa: Benchmarking vision-language models on multi-chart problems. *arXiv preprint arXiv:2410.14179*, 2024.

A APPENDIX

A.1 LLM USAGE STATEMENT

LLMs were used in the work as general purpose writing aid (e.g. to polish grammar and phrasing) and to assist with literature search. All substantive research ideation, experiments and analysis has been conducted by the authors.

A.2 LIMITATIONS

Our work focuses on VLM evaluations and do not assess model fine-tuning. While such approaches might yield stronger results, they diverge from our goal of studying general purpose VLMs for dense level understanding. For dataset construction despite availability of chart datasets with more sophisticated real-world chart examples, we selected the ChartX Xia et al. (2024) dataset because it provides precise chart information in form of csv data and plotting code which is essential for generating precise ground truth values for the evaluation of dense grounding and alignment.

A.3 DATASET CONSTRUCTION

Algorithm 1: ChartAB Dataset Construction: *Data Grounding and Alignment* Subset

---

Input: Source dataset $\mathbf{D}_{\text{ChartX}} = \{(T_i, S_i)\}_{i=1}^N$ from ChartX (Xia et al., 2024), where T_i is a CSV table and S_i is the corresponding plotting script, N is number of instances; Number of cells to modify $k \in \{1, 2, 3\}$; Scaling range $[\alpha_{\min}, \alpha_{\max}]$.

Output: Constructed dataset $\mathbf{D}_{\text{ChartAB}}^{(\text{data})} = \{(x_i, x_i', y_i^g, y_i^a)\}_{i=1}^M$, where x_i, x_i' are chart images, y_i^g is the grounding label, and y_i^a is the alignment label, M is number of instances.

foreach $(T, S) \in \mathbf{D}_{ChartX}$ **do**

  Parse table T to obtain a set of all cells $C = \{(r, c, v_{r,c})\}$, where r and c denote cell's row label and column label respectively, and $v_{r,c}$ the corresponding cell value;
  Identify candidate cells $C' \subseteq C$ with unique values;
  if $|C'| < k$ **then**
       skip this chart;

  Sample k cells $\{(r_i, c_i, v_{r_i,c_i})\}_{i=1}^k$ from C';
  Sample scaling factors $\{\alpha_i\}_{i=1}^k$ from scaling range $[\alpha_{\min}, \alpha_{\max}]$;
  Initialize $T' \leftarrow T$ and $S' \leftarrow S$;
  foreach $(r, c, v_{r,c}) \in C'$ **do**
       Compute modified value $v_{r,c}' = \alpha_i \cdot \mu_c$, where μ_c is the mean of cells in column c;
       if *not (unique match of $v_{r,c}$ in S)* **then**
           skip this chart;
       Replace $v_{r,c}$ with $v_{r,c}'$ in T' and S' ;

  Execute S and S' to generate chart images x and x';
  if *x and x' generation succeed* **then**
       Create instance (x, x', y^g, y^a) where $y^g = (T, T')$ and $y^a = \{(r_i, c_i, v_{r_i,c_i}, v_{r_i,c_i}')\}_{i=1}^k$;
       Append (x, x', y^g, y^a) to $\mathbf{D}_{\text{ChartAB}}^{(\text{data})}$;

We used ChartX dataset Xia et al. (2024) as source dataset for our ChartAlignBench curation. ChartX contains plotting-code and csv data-table for the chart with extremely high level of precision thus offering the flexibility for performing finer-level changes along with ground-truth generation capabilities. It contains diverse chart types of varying complexities, and chart data from multiple domains. Hence enabling analysis across charts of varying difficulties.

We utilize *perturbations* for generating fine-grained variations for given chart thus helping build dense-alignment pairs. Chart's plotting-code is perturbed for precise data or attribute changes based on rigorous formatting check using regex-based search and replace, resulting in chart image generation from code execution.

Algorithm 2: ChartAB Dataset Construction: *Attribute Grounding and Alignment* Subset

---

Input: Source dataset $\mathbf{D}_{\text{ChartX}} = \{S_i\}_{i=1}^N$ from ChartX (Xia et al., 2024), where S_i is the plotting script, N is number of instances; Set of attribute types $B = \{\text{color}, \text{legend}, \text{text style}\}$.

Output: Constructed dataset $\mathbf{D}_{\text{ChartAB}}^{(\text{attribute})} = \{(x_i, x_i', b_i, y_i^g, y_i^a)\}_{i=1}^M$, where x_i, x_i' are chart images, $b_i \in B$ is the attribute type, y_i^g is the grounding label, y_i^a is the alignment label, M is number of instances.

foreach $(T, S) \in \mathbf{D}_{\text{ChartX}}$ **do**
> Parse plotting script S using regex to detect plot attributes;
> color_list $\leftarrow$ locate *unique* color array in S, corresponding to visual encodings (e.g., bars/lines/boxes);
> legend_position $\leftarrow$ extract position parameter from legend(..., loc=·) in S;
> text_style $\leftarrow$ parse rcParams for size, weight, and font family for regions (*title, legend, axes labels, axes ticks*);
> Collect detected attributes {color_list, legend_position, text_style};
> **if** *any attribute value is undefined or ambiguous* **then**
>> **skip this chart**;
>
> // Generate modified versions for each attribute type
> **foreach** *attribute type* $b \in B$ **do**
>> Initialize $S' \leftarrow S$, $y^g \leftarrow \emptyset$, and $y^a \leftarrow \emptyset$;
>> **if** $b = color$ **then**
>>> Sample new color list color_list$'$ by randomly replacing a subset of colors;
>>> Replace color array in S' with color_list$'$;
>>> $y^g \leftarrow (\text{color\_list}, \text{color\_list}')$;
>>> changed_colors $\leftarrow \{(c_{\text{old}}, c_{\text{new}}) \mid c_{\text{old}} \neq c_{\text{new}}\}$;
>>> $y^a \leftarrow \{\text{"type"}: \text{"color"}, \text{"changed"}: \text{changed\_colors}\}$;
>>
>> **else if** $b = legend$ **then**
>>> Sample new legend position legend_position$' \in \{\text{'upper left'}, \text{'upper right'}, \dots\}$;
>>> Replace loc parameter in S' with legend_position$'$;
>>> $y^g \leftarrow (\text{legend\_position}, \text{legend\_position}')$;
>>> $y^a \leftarrow \{\text{"type"}: \text{"legend"}, \text{"changed"}: \text{legend\_position}'\}$;
>>
>> **else if** $b = text\ style$ **then**
>>> Sample new text style parameters text_style$'$ (font size, weight, or family);
>>> Update rcParams in S' with text_style$'$;
>>> $y^g \leftarrow (\text{text\_style}, \text{text\_style}')$;
>>> changed_fields $\leftarrow \{(k, v_{\text{old}}, v_{\text{new}}) \mid \text{text\_style}[k] \neq \text{text\_style}'[k]\}$;
>>> $y^a \leftarrow \{\text{"type"}: \text{"text style"}, \text{"changed"}: \text{changed\_fields}\}$;
>>
>> Execute S' to generate modified chart image x';
>> **if** x' *generation succeeds* **then**
>>> Create instance (x, x', b, y^g, y^a);
>>> Append (x, x', b, y^g, y^a) to $\mathbf{D}_{\text{ChartAB}}^{(\text{attribute})}$;

The csv availability and attribute information enable accurate ground-truth generation. Generated pairs for data alignment and attribute alignment include randomly assigned changes, and robustness sets include diverse attribute values for meticulous and unbiased evaluation.

The algorithmic description for generating chart pairs for *Data Grounding & Alignment* 1, *Attribute Grounding & Alignment* 2, *Robustness* 3 describe the process in detail.

A.4 A TWO-STAGE EVALUATION PIPELINE DETAILS

We utilize natural-language based instructions for zero-shot inference to enable simple execution with minimal task specific nuances for strong generalization across various models.

Algorithm 3: `ChartAB` Dataset Construction: *Robustness Set Generation*

Input: Source dataset $\mathbf{D}_{\text{ChartX}} = \{(T_i, S_i)\}_{i=1}^N$, where T_i is a CSV table and S_i is the corresponding plotting script, N is number of instances; Number of cells to modify $k \in \{1, 2, 3\}$; Scaling range $[\alpha_{\min}, \alpha_{\max}]$; Visual variations per instance $d = 5$; Set of attribute types: $B = \{\text{color}, \text{legend}, \text{text style}\}$.

Output: $\mathbf{D}_{\text{ChartAB}}^{(\text{robust})} = \{\{(x_i^{(j)}, x_i^{'(j)})\}_{j=1}^d, y_i^g, y_i^a, at_i\}_{i=1}^M$ where $x_i^{(j)}, x_i^{'(j)}$ are chart images for variation j, $b_i \in B$ is the attribute type being varied, y_i^g is the grounding label, y_i^a is the alignment label.

foreach $b \in B$ **do**

    foreach $(T, S) \in \mathbf{D}_{\textit{ChartX}}$ **do**

        `// Apply data modification (Algorithm 1)`

        Parse T to extract cells $\{(r, c, v_{r,c})\}$;

        identify unique-value cells C' **if** $|C'| >= k$;

        Sample k cells $\{(r_i, c_i)\}_{i=1}^k$ from C' and scaling factors $\{\alpha_i\}_{i=1}^k$ from $[\alpha_{\min}, \alpha_{\max}]$;

        Create modified table T' and script S' by replacing v_{r_i,c_i} with $v'_{r_i,c_i} = \alpha_i \cdot \mu_{c_i}$;

        if *any v_{r_i,c_i} has non-unique match in S* **then**

            skip this chart

        Set $y^g \leftarrow (T, T')$ and $y^a \leftarrow \{(r_i, c_i, v_{r_i,c_i}, v'_{r_i,c_i})\}_{i=1}^k$;

        `// Generate base pair and visual variations`

        Execute S and S' to generate base charts $x^{(0)}$ and $x'^{(0)}$;

        if *generation fails* **then**

            skip this chart

        Initialize $\mathcal{P} \leftarrow \emptyset$;

        for $j = 1$ **to** v **do**

            Sample variation Δ_j for attribute b (color/legend/text style);

            Apply Δ_j to both S and S' to create S_j and S'_j;

            Execute S_j and S'_j to generate $x^{(j)}$ and $x'^{(j)}$;

            if *generation succeeds* **then**

                Add $(x^{(j)}, x'^{(j)})$ to $\mathcal{P}$

        if $|\mathcal{P}| = v$ **then**

            Append $\{\{(x^{(j)}, x'^{(j)})\}_{j=1}^d, y^g, y^a, at\}$ to $\mathbf{D}_{\text{ChartAB}}^{(\text{robust})}$;

VLM outputs follow *JSON based formatting* due to precise nature of the key-value structure which is essential for element specific information serialization for finer-analysis, along with flexibility for variations in completion of grounding and fine grained analysis. The alignment JSON contains finer level attributes for which the charts differ, and the values for corresponding attribute in the two charts. E.g. for data alignment (as shown in Fig. 4) the finer level attributes changed between the charts i.e. cells are identified by their row & column header, along with its values in the chart pairs, i.e. value in chart 1 & value in chart 2 respectively. Evaluation of attribute alignment tasks follow the same pipeline, as illustrated in Figure 15 for color alignment, Figure 16 for text-style alignment, Figure 17 for legend alignment.

A.5 EVALUATION METRICS

A.5.1 DENSE ALIGNMENT

We evaluate dense alignment performance across four task categories: *data alignment* (subtasks: 1-cell/2-cell/3-cell), *color alignment*, *text style alignment*, and *legend alignment*. Performance on the first three tasks is evaluated by a key-value alignment score, which assess the capability to identify the different elements (keys) between two charts and their associated values. In contrast, legend alignment score mainly focuses on identifying the different positions of legends in two charts (values only) because the key is unique and fixed. Table 2 summarizes the keys and values of each type of elements as well as the notations of their dense alignment scores.

Key-Value Alignment Score. For data, color, and text style alignment tasks, we define *elements* as the atomic units that may differ across chart pairs. Each element is characterized by two components:

- **Key**: A textual identifier that uniquely specifies the element within the chart.
- **Value**: The content or attribute value of the element in each chart of the pair.

The key serves to locate and identify different elements, while the values capture their corresponding data or content. We define the alignment score s_{align} on a chart pair (x, x') as:

$$s_{\text{align}}(x, x') = s_{\text{key}} + s_{\text{value}} \tag{1}$$

where $s_{\text{key}} \in [0, 1]$ measures the key identification and $s_{\text{value}} \in [0, 1]$ measures the precision of predicted values. We rescale $s_{\text{align}}(x, x')$ to $[0, 10]$ for better interpretability. We will apply superscripts, e.g., $s_{\text{align}}^{(\text{data})}(x, x')$, to distinguish different task categorie, as shown in Table 2.

Key Identification Score s_{key} evaluates whether the model correctly identifies different elements between two charts. Let $K_{\text{gt}} = \{k_1, \ldots, k_n\}$ be the set of ground truth keys and $K_{\text{pred}} = \{\hat{k}_1, \ldots, \hat{k}_m\}$ be the set of predicted keys. We perform key matching between K_{gt} and K_{pred} using task-specific criteria: (1) for data and color alignment, we use Levenshtein distance with threshold $\tau = 0.5$ to account for the high lexical diversity of real-world named entities (Cohen et al., 2003) and tabular headers (Zhang et al., 2019); (2) for text style alignment, we require exact matches since the keys are predefined and region-characteristic. Let $K_{\text{valid}} = K_{\text{pred}} \cap_\tau K_{\text{gt}}$ denote the set of valid predicted keys, where $\cap_\tau$ represents the fuzzy intersection operator. We compute the following F1 score as s_{key}:

$$p_{\text{key}} = \frac{|K_{\text{valid}}|}{|K_{\text{pred}}|}, \quad r_{\text{key}} = \frac{|K_{\text{valid}}|}{|K_{\text{gt}}|}, \quad s_{\text{key}} = \frac{2 \cdot p_{\text{key}} \cdot r_{\text{key}}}{p_{\text{key}} + r_{\text{key}}} \tag{2}$$

Precision of Predicted Values s_{value}. For each valid predicted element $k \in K_{\text{valid}}$, we measure the precision of its predicted values in both charts. Let (v_k, v'_k) and $(\hat{v}_k, \hat{v}'_k)$ denote the ground truth and predicted values in charts x and x' respectively. The precision of predicted values is defined as

$$s_{\text{value}} = \frac{1}{2|K_{\text{valid}}|} \sum_{k \in K_{\text{valid}}} \left(\rho(v_k, \hat{v}_k) + \rho(v'_k, \hat{v}'_k) \right) \tag{3}$$

where $\rho(\cdot, \cdot) \in [0, 1]$ denotes a task-specific value matching function: it performs exact matching for categorical attributes (e.g., text weight/font), $\rho(v, \hat{v}) = 1 - \|v - \hat{v}\|_2$ for color attributes (with $\|v - \hat{v}\|_2$ denoting the normalized RGB distance), and $\rho(v, \hat{v}) = 1 - \min(|v - \hat{v}|/|v|, 1)$ for numerical attributes (e.g., data values or text size).

Task	Score	Key	Value
Data Alignment	$s_{\text{align}}^{(data)}(x, x')$	Row and column labels (e.g., "John, Salary")	Numerical value (float/int)
Color Alignment	$s_{\text{align}}^{(color)}(x, x')$	Series/category label (e.g., "Product A")	Hex color code (e.g., "#FF5733")
Text Style Alignment Alignment	$s_{\text{align}}^{(text-style)}(x, x')$	Region-characteristic pair (e.g., "title-size")	Style attribute value (size: int, weight/family: categorical)
Legend Alignment	$s_{\text{align}}^{(legend)}(x, x')$	Position (implicit)	3X3 grid (center, upper, ...)

Table 2: Chart elements' keys, values, and scores in the four categories of dense alignment tasks. For data, color, and text style alignment, fuzzy matching (Levenshtein distance $\tau = 0.5$) or exact matching is used to evaluate the key identification, while the precision of associated values are evaluated using $\rho(\cdot, \cdot)$. Legend alignment score is defined by spatial distance between the values of legend positions.

Legend Alignment Score. Unlike the above three alignment tasks, legend alignment only focuses on one unique key, i.e., the legend position, so the legend alignment score is defined as the spatial proximity between the ground truth and model-detected positions. We discretize the chart into a 3×3 grid and measure the Manhattan distance between predicted and ground truth legend positions. The legend alignment score is defined by

$$s_{\text{align}}^{(\text{legend})}(x, x') = 1 - \frac{1}{10} \cdot (d_{\text{Manhattan}}(pos, \hat{pos}) + d_{\text{Manhattan}}(pos', \hat{pos'})) \tag{4}$$

where $\hat{pos}$ and pos are the predicted and ground truth positions, and $d_{\text{Manhattan}}(\cdot, \cdot) \in [0, 5]$ is the Manhattan distance. We normalize $s_{\text{align}}^{(\text{legend})}(x, x')$ to $[0, 10]$ for better interpretability.

For each chart type, we report the averaged alignment scores over all the chart pairs belonging to that chart type.

A.5.2   ROBUSTNESS

We evaluate the robustness of data alignment performance to the variations of visual attributes. For a chart pair (x, x') differing in a 1-3 data cells, we define robustness $r(x, x')$ as the reciprocal of the standard deviation $\sigma(\cdot)$ of alignment scores across d visual variations:

$$r(x, x') = \frac{1}{1 + \sigma\left(\left\{s_{\text{align}}^{(\text{data})}(x^{(j)}, x'^{(j)})\right\}_{i=1}^{d}\right)} \tag{5}$$

where $(x^{(j)}, x'^{(j)}), \ldots, (x^{(d)}, x'^{(d)})$ are the d visually-varied versions of the same chart pair (x, x'), and $s_{\text{align}}^{(\text{data})}$ denotes the data alignment score. Higher $r(x, x')$ indicates more consistent data alignment performance across different visual variations. We compute robustness separately for each attribute $a \in \{\text{color}, \text{legend}, \text{text style}\}$. For each chart type, we report the robustness score averaged over all the chart pairs belonging to that chart type.

A.6   ADDITIONAL EXPERIMENTAL DETAILS

A.6.1   VLM SELECTION

We evaluate a diverse suite of *open-source VLMs* from following families: Phi-3.5 vision-instruct Abdin et al. (2024), InternVL-2.5 (8B) Chen et al. (2024), LLaVA-1.6 Mistral (7B) Liu et al. (2023a), QWEN-2.5 VL (8B) Bai et al. (2025). These models constitute among most widely used VLMs, and have a long timeline of continuous evolution with each released version. The set encompasses the top-performed VLMs in various chart benchmarks (CharXiv Wang et al. (2024b), ChartQAPro Masry et al. (2025), SCI-CQA Li & Tajbakhsh (2023), MultiChartQA Zhu et al. (2024), discussed in 2).

Our choice of *proprietary VLM* is based on CharXiv Wang et al. (2024b) leaderboard as its tasks/questions require dense-level grounding. For example, CharXiv tasks need to identify axes ticks by positions and their value enumerartion, grid-lines count and intersections, integral (area comparison of regions) and slope (rate of increase/decrease) in line charts. And GPT-4o Hurst et al. (2024) is the best performing proprietary in the CharXiv paper.

Among *chart-specialized VLMs*, we evaluate TinyChart Zhang et al. (2024b) & ChartGemma Masry et al. (2024) models. However, due to their task-specific training (discussed in 2), these models show collapse of instruction following capabilities and fail to output required JSON format needed for evaluation. Below are a few examples of the outputs.

JSON output: Data alignment (1 cell) by ChartGemma and TinyChart models using 1-stage stitched-charts (i.e chart pair stacked as single image) evaluation.

```
REQUIRED FORMAT (specified in prompt instructions):-
{"row name": <row name of the cell>, "column name": <column name of the cell>,
 "value in chart 1": <value in first chart of the pair>, "value in chart 2":
 <value in second chart of the pair>}


EXAMPLE:-
```

```
{"row name": "Production A (million units)", "column name": "2021",
 "value in chart 1": 35, "value in chart 2": 30}

CHARTGEMMA OUTPUT (abnormal valued JSON which is inconsistent with required format):-
{"row name": "sample row", "column name": "sample column",
 "value in chart 1": Infinity, "value in chart 2": Infinity}

TINYCHART OUTPUT (abnormal list instead of JSON):-
["Production A (million units)", "Production B (million units)",
 "Production C (million units)" ..... "Production Z (million units)"]
```

A.6.2 ABLATIONS

Type	Approach	Bar	Bar #	3D Bar	Line	Line #	Radar	Rose	Box	Multi-Axes
1-stage	Multi-chart	2.6	4.5	1.9	2.9	3.0	1.1	0.1	0.9	0.8
	Stitched-chart	2.1	2.2	0.8	1.9	0.9	0.5	0.1	0.1	0.4
2-stage	Ours	4.7	7.0	1.7	5.4	5.9	1.0	0.1	0.4	0.7

Table 3: **Ablation study of 1-stage vs. 2-stage evaluations** on data alignment (one cell change) task. Mean scores across nine chart types show that our 2-stage evaluation reflects VLMs' greatest potential on chart alignment.
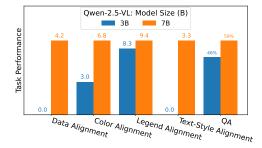
We performed ablation experiments to vigorously compare differing approaches to our 2-stage approach.

The ablation experiments aimed to thoroughly compare single-stage based alignment approaches for performing multi-image reasoning vis-a-vis our two-stage approach. The ablation techniques:-

(1) *stitched-charts* inference: The chart-pair images are vertically concatenated resulting in a single image of stitched chart-pairs which undergo single-stage inference.

(2) *multi-image* inference: The VLM inputs multiple images, and contextualizes output based on the input images with aim of better understanding across of finer-level alignment in multi-image reasoning.

The ablation experiments evaluated the Phi-3.5 model's performance on the data alignment task. As shown in Table 3, the single-stage approach underperformed compared to our proposed two-stage method, reaffirming the effectiveness of intermediate grounding for reasoning in the alignment task, helping to focus more precisely on localized relationships between visual and textual elements. In contrast, the single-stage approach struggles to capture these fine-grained correspondences due to information loss during joint encoding and limited cross-attention resolution. Despite continued progress in multi-modal training, current VLMs still face challenges in detailed reasoning, and our results highlight how decomposing complex tasks like our multi-chart dense alignment into modular stages can substantially mitigate these limitations.
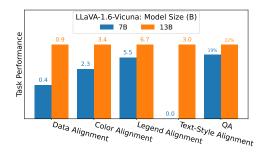
A.7 ADDITIONAL FINDING & INSIGHTS

Figure 11: Task performances for different sizes of Qwen-2.5-VL and LlaVa-Vicuna-1.6.

(a) Legend Alignment
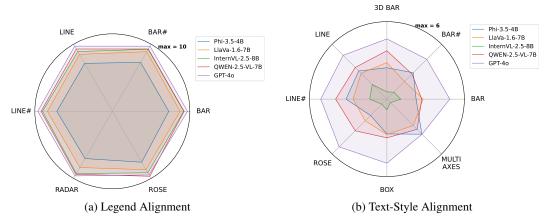
(b) Text-Style Alignment

Figure 12: (a) **Legend alignment** of legend positions. Phi-3.5 performs the worst while GPT-4o is best. Related discussion in Finding 1&2. (b) **Text-style alignment** (size, weight, font). Worst: InternVl-2.5-8B, Best: GPT-4o. Discussion in Finding 1&4.

Finding 7

VLMs' data grounding and alignment are more robust to color variations than changes in legend positions and text styles.

Fig. 13 shows that robustness is the worst under text-style variations and the best under color variations. In the visualizations of data, colors are used to discretize, categorize, and measure chart constituents. As long as their colors are distinguishable, color variations will not affect the data grounding. In contrast, the text styles and legends provide critical information about the data via ticks, labels, and legend items. Moreover, changing legend position may lead to position changes and occlusion of other chart elements. Hence, their variations have a greater impact on the data grounding/alignment performance.
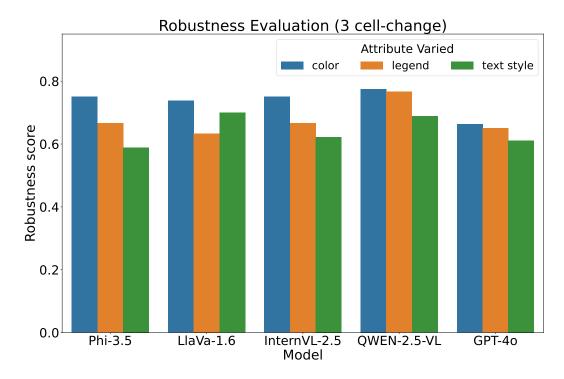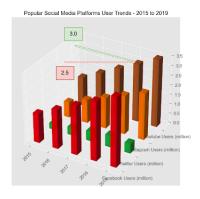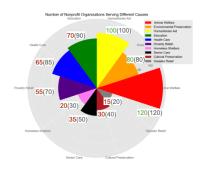

Figure 13: **VLMs' Robustness of data alignment (3-cell change) to variations in color, legend, and text-style.** VLMs show better robustness to color changes than text-style changes. QWEN-2.5-VL outperforms the other four VLMs on robustness. More discussion can be found below Finding 6.

(a) Depth estimation in 3D bar charts

(b) Text vs. non-text cues for value scaling in rose charts.

Figure 14: **VLMs' spatial understanding is poor on complex charts.** More discussion is provided below Finding 7.

Finding 8

VLMs' spatial understanding capability affects several important chart understanding skills.

Chart understanding usually requires an accurate mapping between spatial relationships and the corresponding numerical values to be visualized.

- *Depth understanding*: Despite the high-level similarity between 3D bar charts and (2D) bar charts, as shown in Fig 5, the data alignment performance is much poorer on 3D bar charts due to the lack of depth understanding, which affects the measurement of scales and values along axes in the 3D space.

- *Text vs non-text cues*: Rose charts are extended from bar charts by allowing more polar coordinates with scale differences in radial forms. However, Fig. 14b reveals a great difference between the two on data alignment performance. This is due to fewer text cues (e.g., axes ticks) in rose charts, where non-text cues such as grid lines cannot be fully leveraged.

- *Better performance on numbered charts*: numbered bar and line charts explicitly place the data values in the charts, hence facilitating VLMs to extract the data easily without precise measurements of the visual elements. Hence, as shown in Fig. 5, numbered bar/line charts usually enjoy better performance.
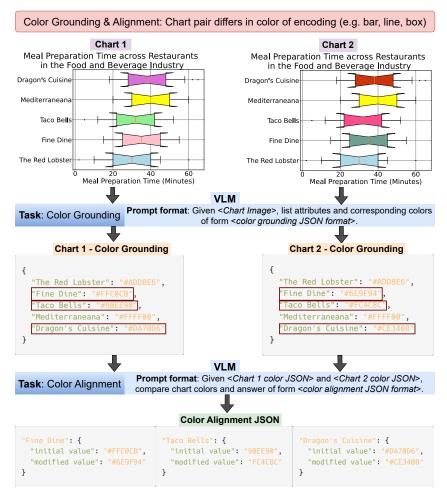
Figure 15: **Two-Stage Evaluation Pipeline for *Color Grounding & Alignment* in `ChartAB`.** The first stage focuses on grounding the color for visual encodings in each chart, while the second stage focuses on alignment, which aims to evaluate the colors for visual encodings and output a JSON file listing the visual encodings which differ in color values between the chart pair.
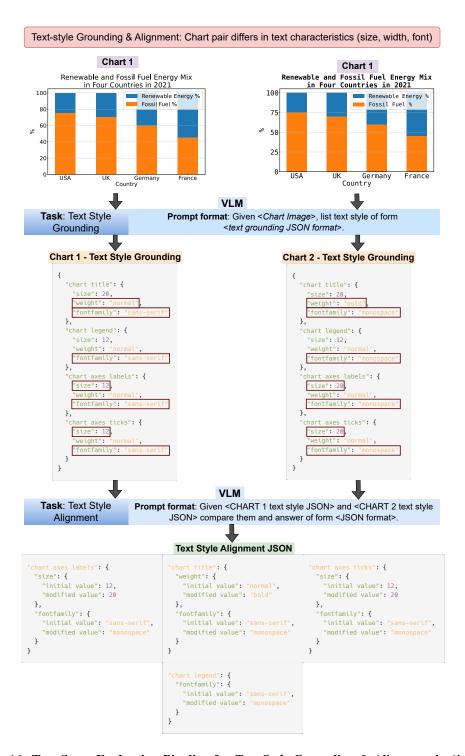
Figure 16: **Two-Stage Evaluation Pipeline for *Text Style Grounding & Alignment* in `ChartAB`.** The first stage focuses on grounding the text characteristics for the four chart regions: title, legend, axes labels, axes ticks. These characteristics are textual size, weight (lightness/boldness), and font family (e.g., Times New Roman). The second stage focuses on alignment, which aims to evaluate the grounded text characteristics and output a JSON file listing the characteristics for each region which differ between the chart pair.
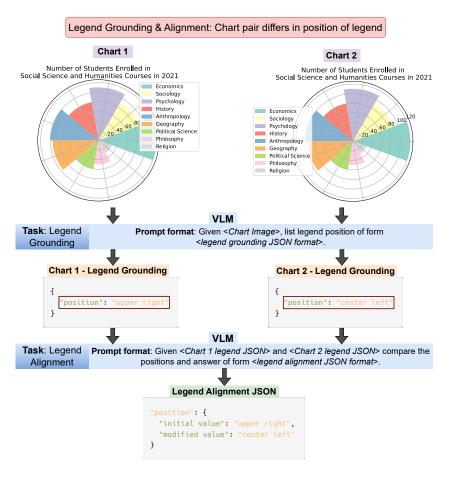
Figure 17: **Two-Stage Evaluation Pipeline for *Legend Grounding & Alignment* in `ChartAB`.** The first stage focuses on grounding the legend position in each chart, while the second stage focuses on alignment, which aims to determine the difference in the position and output the JSON file listing the difference.