THEORY-DRIVEN CONCEPTUALIZATION AND IMPLEMENTATION OF EVALUATION METRICS FOR CHART EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Charts can be visually similar yet communicate different information, while visually different charts may support the same interpretation. This makes chart comparison fundamentally different from conventional image similarity and poses considerable evaluation challenges in tasks such as chart editing and chart generation. We define this as a matter of *functional equivalence*: whether a candidate chart preserves the information and inferences supported by a reference chart. To support the study of functional equivalence, we introduce CHARTFUN, a benchmark of human judgments on the functional equivalence of 2,400 model-generated charts against rule-based targets. CHARTFUN spans five chart types and six image- and code-based chart-generation models. Our analysis reveals substantial variation across models: only 60.5% of generated charts are judged functionally equivalent to their rule-based targets overall, despite many outputs being visually similar. More importantly, correlation studies reveal that existing evaluation metrics used in chart-generation studies show limited agreement with human judgments of functional equivalence. We explore three new evaluation approaches grounded in prior theory on data visualization: a vision-language model (VLM) judge, a structured theory-driven comparison pipeline, and a lightweight learned evaluator. Our best-performing method, the VLM Judge, reaches a Pearson correlation of 0.60. Our work opens new avenues for evaluating chart generation and editing through the theory-grounded framework of functional equivalence.

1 INTRODUCTION

Charts communicate information through the joint function of multiple visual elements, which viewers read and interpret through perceptual and cognitive operations (Cleveland & McGill, 1984; Pinker, 1990; Ratwani & Trafton, 2008). Visually different representations—e.g., a grouped bar chart and a heatmap—can encode the same underlying information, while seemingly minor visual changes—such as reassigning a tick value or reducing a bar height—can substantially alter the information encoded in a chart. Mackinlay (1986) dubs this *expressiveness*: the requirement that a data visualization encodes all and only the specified information. On the decoding side (the viewer perspective), equally expressive visualizations can differ in terms of *effectiveness*, i.e. how well they support the accurate perceptual decoding of visual variables (Mackinlay, 1986; Cleveland & McGill, 1984). Plotting values in the range of 90.4 to 90.9 on a 0–100 axis, for example, impedes the accurate reading of the values. Beyond surface-level perception, viewers draw *inferences* from data visualizations. Deceptive design tactics, such as truncated axes and arbitrary sizing, can cause viewers to misinterpret the facts (Lauer & O'Brien, 2020). From an algebraic design perspective, a visualization is misleading when visually salient changes correspond to unimportant changes in the underlying data (Kindlmann & Scheidegger, 2014). Viewers' interpretations are further shaped by chart format, familiarity with the content, and graph literacy (Shah & Freedman, 2011).

Despite a plethora of research on graph perception and comprehension, current approaches to evaluating AI-generated and AI-edited charts lack theoretical grounding. At the lowest level, generic image-similarity metrics such as SSIM, PSNR, LPIPS, and CLIP similarity (Wang et al., 2004a;b; Zhang et al., 2018; Radford et al., 2021) are used to compare rendered candidates and references at the pixel or feature level (Li et al., 2025). More structured approaches match predefined chart

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

components in the generated plotting code, such as chart type, layout, text, and color (Yang et al., 2025), or compare textual elements extracted from rendered charts using OCR (Wu et al., 2025). Many benchmarks additionally employ multimodal LLMs to judge either holistic visual similarity or criteria such as instruction following, data faithfulness, layout, style, and clarity (Xia et al., 2025; Yang et al., 2025; Wu et al., 2025; Yang et al., 2024; Li et al., 2025; Zhao et al., 2025). At the most structured end, ChartEval parses reference and candidate charts into scene graphs and measures their correctness, completeness, hallucination, and omission (Goswami et al., 2025). These criteria are closer to classical natural language generation (NLG) evaluation (Dušek & Kasner, 2020) than any theory of data visualization. There are currently no approaches which provide a common account of which discrepancies alter the information encoded by a chart, which affect how accurately that information can be perceived, and which change the inferences available to a viewer, or that attempt to distinguish the severity of different discrepancies, at any of these levels.

The human validation of existing metrics, where available, is also limited. Often, human-based studies employ different judgment criteria and narrow pools of systems (Wu et al., 2025; Yang et al., 2025; 2024), and some use the same model both to generate and to evaluate the charts (Wu et al., 2025; Yang et al., 2025). Although ChartEval (Goswami et al., 2025) conducts a broader study across multiple datasets and models, its human judgments are organized around the same NLG categories defined by the metric itself, the exact code for the metric described in the paper is not available, and the human ratings are not released as a reusable benchmark. We address these limitations with an extensive overview of the relevant literature on graph perception and comprehension, a position on how that should be translated into operational concepts and evaluation metrics for chart generation, and a concrete instantiation of a metric grounded in one of these key concepts: functional equivalence.

We posit that for the purposes of chart generation evaluation, encoding-level expressiveness and decoding-level effectiveness are inseparable, since charts are always produced to be consumed by a viewer. Recognizing that existing terminology from the data visualization theory mostly concerns a chart and its underlying facts, while reference-based evaluation concerns two juxtaposed charts, we introduce a new, very transparent term: functional equivalence, to mean that a candidate chart encodes all and only the information specified in a reference chart (equivalent to expressiveness), and does so in an effective manner, i.e. with ease of perceptual decodability. We substantiate functional equivalence with a taxonomy of 36 discrepancies (Lo et al., 2022) organized in 5 tiers of severity.

To assess functional equivalence, we construct a human-validated benchmark comprising 2,400 charts generated by six frontier models: four models for code generation and two models for image generation. Each chart comparison is annotated independently by three human judges. We evaluate standard visual similarity metrics and chart-specific metrics on this benchmark, alongside our own instantiations of established approaches, including embedding-based similarity, structured rule-based scoring, and VLM-as-a-judge, the latter two approaches being theory-grounded by design.

Our contributions are as follows:

1. **Functional equivalence.** We frame chart equivalence across encoding, decoding, and interpretation, and formalize *functional equivalence* as whether two charts preserve the same information and support the same principal viewer inferences.
2. **The CHARTFUN benchmark.** We introduce a benchmark of 2,400 model-generated charts spanning six chart-generation models, each independently rated by three human judges.
3. **Evaluation of functional equivalence.** We benchmark general-purpose image-similarity and chart-specific metrics against human judgments, and introduce three new evaluators, based on image similarity, structured chart comparison, and a theory-guided VLM judge.

2 RELATED WORK

Key works in the space of chart generation and chart editing rely on non-chart-specific metrics and loosely defined LLM judges. ChartX (Xia et al., 2025) includes a task on chart redrawing where a rendered chart has to be mapped to Python code. GPT-score is used to assess the generated code on a scale of 0 to 5. ChartMimic (Yang et al., 2025) and Plot2Code (Wu et al., 2025) address both chart reconstruction and chart editing, again as image-to-code generation. Both works evaluate the rendered output using GPT-4o/4V image similarity in overall appearance, colors, shapes, positions,

and other high- and low-level visual elements; ChartMimic further uses a set of F1 scores over predefined key elements of the plotting code (e.g., subplot layout, color), while Plot2Code measures OCR-based text-match ratio. MatPlotBench (Yang et al., 2024) focuses on instruction-based chart generation and evaluates using an image similarity score from GPT-4V. FigEdit (Li et al., 2025) studies a range of chart editing subtasks, where the output can be either a generated image or an executable chart specification, rendered into an image for the purposes of evaluation. A wide range of visual similarity metrics are used to assess rendered outputs (SSIM (Wang et al., 2004a), PSNR (Wang et al., 2004b), LPIPS (Zhang et al., 2018), CLIP (Radford et al., 2021) similarity, OCR similarity), while an LLM judge is used to score instruction following, content preservation, and visual quality. ChartEdit (Zhao et al., 2025) focuses on chart editing through code generation, with evaluation on the level of both code and rendered image, using GPT-4o to score code accuracy and completeness, and visual aspects such as chart type, data faithfulness, layout and clarity. ChartEval (Goswami et al., 2025) parses reference and candidate charts into scene graphs of chart elements, attributes, and relations. Correctness and completeness are computed as precision- and recall-like Graph-BERTScore matches, while hallucination and omission measure candidate-only and reference-only graph elements, respectively.

We identify two key limitations with the metrics used in these works. (1) They are not grounded in the theory of graph perception and comprehension. Evaluation based on visual similarity completely disregards the basic premise of expressiveness, while evaluation protocols that define some evaluation criteria around e.g. data faithfulness, layout and clarity, do so with no reference to visualization theory, with the risk of overlooking key aspects and conflating criteria across different tiers of severity when it comes to the faithful encoding of information. (2) Existing evaluation metrics are scarcely assessed against human judgments with inconsistent results. Plot2Code (Wu et al., 2025) measures correlation between their GPT-4V visual-similarity scores and 100 human raters on 100 charts generated by the same model (GPT-4V) and observe a Pearson correlation of 0.61. MatPlotBench (Yang et al., 2024) evaluates the same judge, GPT-4V, on an unspecified number of human judgments, and report a Pearson correlation of 0.836 and 0.876 for two different chart generation models (GPT-3.5 and GPT-4). ChartMimic (Yang et al., 2025) asks three human judges to rate 1,200 charts generated by GPT-4o on a scale of 0 to 100 based on similarity with a ground-truth chart. They observe a Pearson correlation of 0.704 for their GPT-4o visual-similarity score and 0.768 for their code-based multi-criteria score. One can question the validity of human judgment meta-evaluations that use the same model to generate and evaluate charts (Plot2Code and ChartMimic), and the decision to include only a single model in the meta-evaluation pool (all three). Lastly, ChartEval (Goswami et al., 2025) reports the most comprehensive human meta-evaluation, on 4,000 charts, sampled from three datasets, generated by four models and rated by three annotators each. They report a Pearson correlation of 0.45 to 0.89 for the four components of their metric, across different models, on average superior to image- and code-similarity metrics as well as a visual similarity score based on GPT-4o. Unfortunately, none of these four datasets of human judgments were ever made public, which limits the reproducibility of the reported results and the future development of evaluation metrics.

We close these gaps by grounding functional equivalence in theories of graphical perception and comprehension (Section 3), introducing a benchmark of human judgments of functional equivalence (Section 4), and developing theory-grounded metrics evaluated against these judgments (Section 5).

3 Theoretical Framework

Charts communicate data through visual encodings that viewers interpret perceptually and cognitively (Cleveland & McGill, 1984; Pinker, 1990). Cleveland & McGill (1984) show that quantitative judgments depend on visual properties such as position, length, angle, and area. Mackinlay (1986) distinguishes *expressiveness*, which requires a visualization to encode all and only the intended information, from *effectiveness*, which concerns how well the encoding exploits the capabilities of the human visual system. However, preserving the same data does not guarantee the same interpretation. Shah & Freedman (2011) show that presenting identical data as bars or lines can emphasize different relationships and lead viewers to different inferences. Together, these results motivate evaluating charts by the information and inferences they support rather than by visual appearance alone. We use this literature to distinguish three levels of chart communication: *encoding*, concerning which information is represented; *decoding*, concerning how accurately that information can be perceptually

recovered; and *interpretation*, concerning the comparisons and conclusions drawn from it. Figure 1 provides examples of functionally equivalent and non-functionally equivalent comparisons.

Table 1: The 36 chart transformations which characterize functional equivalence.

#	Transformation	Definition
1	Background colour change	Change the chart background colour.
2	Font family change	Change the displayed font family.
3	Consistent colour palette change	Apply a consistent colour palette change to marks and legend.
4	Legend relocation	Move the legend to a different position.
5	Gridline toggle	Add or remove gridlines.
6	Reduced colour discriminability	Use series colors that are difficult to distinguish.
7	Layout adjustment	Change canvas padding or chart layout.
8	Additional minor axis ticks	Add unlabelled minor ticks between existing major ticks.
9	Chart title removal	Remove the chart title.
10/11	X/Y-axis title removal	Remove the x-axis title.
12	Metadata removal	Remove a collection of metadata, such as titles and legend.
13	Axis tick-mark removal	Remove axis tick marks while retaining tick labels.
14	Irregular tick intervals	Introduce an inconsistent interval into regularly spaced ticks.
15/16	Expanded/Truncated axis range	Considerably expand/truncate the displayed quantitative range.
17	Linear-to-log scale change	Replace a linear quantitative axis with a base-10 logarithmic axis.
18	Unequal bar thickness	Change the thickness of selected bars.
19/20	X/Y-axis tick-label removal	Hide all x-axis tick labels.
21	Category reordering	Reorder categories while keeping their data attached.
22	Grouped-to-stacked bars	Convert grouped bars into stacked bars.
23	Secondary-axis assignment	Move selected series to an independently scaled secondary axis.
24	Bars-to-lines conversion	Replace bars with connected lines.
25	Legend colour reassignment	Swap colour assignments in the legend.
26	Legend label replacement	Replace legend labels.
27	Contradictory value labels	Add numeric labels that contradict the plotted values.
28	Value reassignment	Swap existing values between categories or pie wedges.
29	Data value modification	Change one or more underlying data values.
30	Axis inversion	Reverse a quantitative or ordered axis.
31/32	Data addition/removal	Add/remove series or pie wedges.
33	Legend removal	Remove the legend.
34	Bars-to-points conversion	Replace bars with unconnected points at their value endpoints.
35	Categorical line connection	Connect categorical values with lines to imply continuity.
36	Unrelated title replacement	Replace the chart title with unrelated metadata.

We define *functional equivalence* between a reference chart R and candidate chart C by whether C preserves the information and inferences supported by R. Functional equivalence therefore does not require visual identity. Functional equivalence spans these three levels: a candidate should preserve the encoded information, allow it to be effectively decoded, and support the same principal interpretations as the reference. Kindlmann & Scheidegger (2014) provide a related principle through *representation invariance*. For example, reordering countries in a visualization of the same employment data changes the apparent pattern in a graphical design even though every country's values remain unchanged, while a scatterplot of the same data is unaffected by the ordering. This shows that a change in representation can alter the visual impression without changing the underlying facts.

To characterize such changes systematically, we build primarily on the taxonomy of misleading and uninformative visualizations developed by Lo et al. (2022). From real-world examples, they identify 74 issue types spanning problems in data selection, visual design, plotting, perception, and interpretation. These include misleading axes and scales, inappropriate chart types, ineffective choice of colors, missing chart elements, inconsistent mappings, and misleading text. They further distinguish *distortion*, where a visualization communicates a different message from the data, from *confusion*, where relevant information cannot be effectively recovered. These outcomes do not map one-to-one onto Mackinlay's criteria: incorrect or missing information violates expressiveness, whereas information that is present but difficult or misleading to recover violates effectiveness.

Not all chart transformations are equally severe. Larkin & Simon (1987) distinguish *informational equivalence*, where the information represented by either representation can be inferred from the

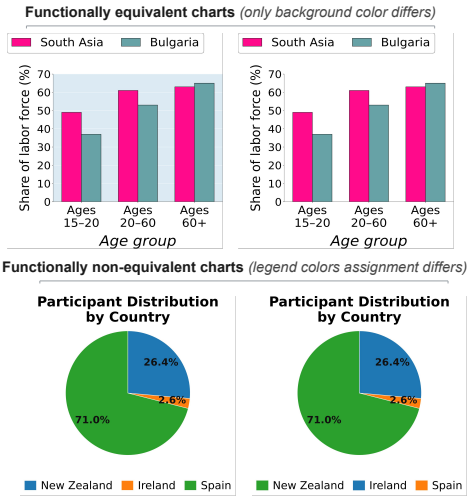

Figure 1: Examples of functionally equivalent and non-equivalent chart pairs.

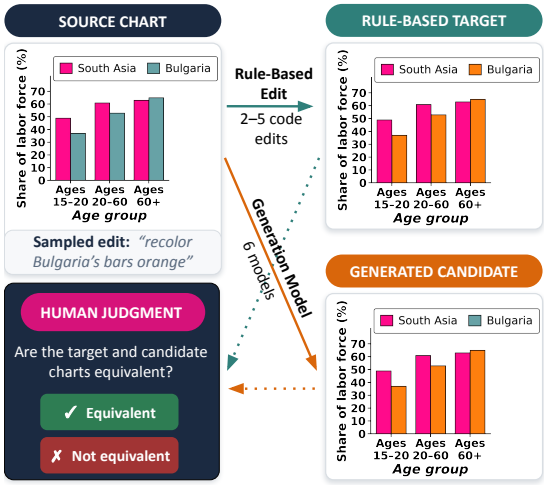

Figure 2: Construction of the human judgement benchmark.

other, from *computational equivalence*, where those inferences can also be made with comparable ease. Pandey et al. (2015) further distinguish *message exaggeration/understatement*, where the underlying relation is preserved but its extent is distorted, from *message reversal*. We use these distinctions to define five severity tiers: cosmetic changes that preserve both informational and computational equivalence (T0); presentation changes that preserve information but make it harder to recover (T1); magnitude distortions that preserve direction and ranking (T2); structural changes that leave the data recoverable but alter the most natural comparison and may produce visualization mirages (McNutt et al., 2020) (T3); and falsifications that encode information absent from the reference or reverse its conclusions, thereby violating expressiveness (Mackinlay, 1986) (T4). In terms of the three levels above, T0 preserves encoding, decoding, and interpretation; T1 primarily affects decoding; T2 and T3 affect interpretation with increasing severity; and T4 changes the encoded information or reverses a principal inference.

We translate this theoretical space into 36 controlled chart transformations. Of these, 24 are direct or close adaptations of issues identified by Lo et al. (2022), including truncated and inverted axes, inappropriate axis ranges, inconsistent tick intervals, dual axes, inappropriate chart encodings, missing titles and legends, ineffective colors, confusing legends, and misleading labels. We specifically chose these 24 as they fit our objective of quantifying functional equivalence between a source and candidate chart. Therefore, the transformation we choose needs to be observable and can be applied to our data visualization types. We introduce the remaining 12 transformations to cover controlled changes to appearance, layout, metadata, and data that are useful for testing functional equivalence. Seven of these 12 are adapted from Yan et al. (2024)'s style and layout transformations. Together, the transformation set includes changes intended to preserve communicated information as well as changes that may obscure or alter it. Table 1 categorizes the proposed transformations.

4 ChartFun

4.1 Benchmark Construction

We construct ChartFun from 400 source charts sampled from PlotQA (Methani et al., 2019), which contains charts (as code and rendered image) derived from real-world sources, including World Bank Open Data, Open Government Data, and the Global Terrorism Database, and comprises horizontal-bar, vertical-bar, line, and dot-line plots. We select 80 charts from each of these four families. We additionally construct 80 pie-chart sources from unused bar-chart annotations with additive or compositional values suitable for a part-to-whole representation. Given this source data, we simulate chart-editing tasks in which a user provides a rendered chart and requests two to five

Table 2: Functional-equivalence rates by generation model and chart type.

Generation model	Mode	H. bars	V. bars	Lines	Marked	Pies	Average
Gemini 3.6 Flash	Code	87.5	85.0	85.0	80.0	87.5	85.0
GPT-5.5	Code	83.8	81.3	83.8	75.0	95.0	83.8
Qwen3.8-27B	Code	52.5	75.0	81.3	66.3	71.3	69.3
GPT Image 2	Image	35.0	42.5	52.5	53.8	86.3	54.0
Gemini 3 Pro Image	Image	33.8	38.8	41.3	42.5	72.5	45.8
InternVL3.5-38B	Code	10.0	23.8	18.8	13.8	58.8	25.0

transformations (sampled from the set in Table 1).[1] Rule-based targets are derived programmatically from PlotQA's underlying code: for each transformation, we implement a corresponding program and apply it to the original code to produce the intended edited chart. Image-generation models are prompted to apply the requested transformations to the input chart, whereas VLMs receive the same prompt along with an instruction to return the edited chart's Python 3/Matplotlib code.

Each evaluation instance contains a rule-based reference, R, and a model-generated candidate C_m. The evaluation compares R with C_m. Figure 2 summarizes the construction of the rule-based references and model-generated candidates. For every applied transformation, we record the information needed to describe that specific operation and insert it into a transformation-specific instruction template. Some transformations require no additional specification, such as "Remove the chart title." Others require exact parameters. The resulting instructions are ordered according to the applied transformation sequence and combined into a single model prompt.

We evaluate six generation models. GPT Image 2 (OpenAI, 2026b) and Gemini 3 Pro Image (Raisinghani, 2025) generate the candidate image directly. GPT-5.5 (OpenAI, 2026a), Gemini 3.6 Flash (Google DeepMind, 2026), Qwen3.8-27B (Qwen Team, 2026), and InternVL3.5-38B (Wang et al., 2025) generate executable plotting code using Matplotlib. For code-generating models, we execute the returned program to obtain the candidate chart. If generation for a source cannot be completed, that source is replaced with another source from the same chart family and the same requested transformations are applied. Across all six models, the 400 sources produce 2,400 rule-target and candidate pairs.

4.2 Human Evaluation

Each of the 2,400 rule-target and candidate pairs is independently evaluated by three qualified Prolific annotators. Participants judge whether the two charts are functionally equivalent by answering whether they show the same data and convey the same meaning. Image order is randomized. Before the main study, participants complete 10 practice comparisons with feedback, followed by a 10-item qualification test without feedback. The qualification examples cover all five chart families and contain both equivalent and non-equivalent pairs. Participants must answer at least 7 of the 10 qualification comparisons correctly to proceed. Overall, 93 qualified participants contributed 7,200 judgments, giving exactly three judgments from distinct annotators for every pair. Final labels are assigned by majority vote. Annotators reached 76.5% raw pairwise agreement, with unanimous judgments on 64.7% of pairs. Fleiss' κ and Krippendorff's α were both 0.51. Under majority voting, 60.5% of all generated candidates were judged functionally equivalent to their rule-based targets.

Human evaluation reveals substantial differences across generation models and chart families. Table 2 reports majority-vote functional equivalence by model and chart family. Performance varies substantially across generators and chart types, with overall equivalence rates ranging from 25.0% to 85.0%. Gemini 3.6 Flash and GPT-5.5 achieve the highest overall functional-equivalence rates at 85.0% and 83.8%, followed by Qwen3.8-27B at 69.3%. GPT Image 2 and Gemini 3 Pro Image reach 54.0% and 45.8%, while InternVL3.5-38B reaches 25.0%. Generation mode alone does not explain these differences, since both the highest and lowest performing models generate code. Chart

[1] With equal distribution between requests for two, three, four, and five transformations.

type also has a substantial effect. The two direct image-generation models perform particularly well on pie charts relative to bar charts, e.g. GPT Image 2 reaches 86.3% on pies compared with 35.0% on horizontal bars and 42.5% on vertical bars. Across all 2,400 candidates, 60.5% are judged functionally equivalent to their rule-based targets.

4.3 AUTOMATIC EVALUATION

The human study provides labels of functional equivalence, however, it is unclear whether existing automatic metrics capture the same distinction. Small changes to a tick, legend binding, or value label can change chart meaning while affecting few pixels, whereas substantial visual changes can leave the conveyed information intact. We therefore compare existing evaluation metrics against the binary majority-vote labels in CHARTFUN. We report Pearson, Spearman, and Kendall correlations between metric scores and the majority-vote labels, together with AUROC. Positive correlation indicates that pairs judged functionally equivalent tend to receive higher metric scores, while AUROC measures how well the metric separates equivalent from non-equivalent pairs. We note that the metrics we consider, including those used for comparison, at best assess expressiveness and effectiveness; none directly evaluates human inference or interpretation.

As seen in Table 3, existing metrics show limited agreement with human judgments of functional equivalence. DreamSim is the strongest general visual-similarity metric, with a Pearson correlation of 0.46 and AUROC of 0.76. CLIP and SigLIP reach Pearson correlations of 0.22 and 0.23 respectively, while SSIM falls to 0.19. Chart-specific metrics do not improve substantially on this result. The ChartEval variants obtain Pearson correlations between 0.14 and 0.22 and AUROC between 0.59 and 0.63. GPTScore performs considerably better, reaching 0.54 Pearson correlation and 0.83 AUROC, but still shows only moderate agreement with the human labels. These results motivate evaluation methods designed specifically for functional equivalence.

5 THEORY-DRIVEN METRICS OF FUNCTIONAL EQUIVALENCE

Measuring functional equivalence is difficult for general visual similarity metrics. Metrics based on pixels or global image representations can be insensitive to small chart elements that carry substantial information. For example, changing a single tick value or legend binding may affect only a small image region while changing the meaning of the chart. We therefore evaluate both general visual similarity metrics and methods designed explicitly for functional equivalence.

5.1 VLM JUDGE

Our first functional-equivalence evaluator uses a VLM (GPT-5) as a reference-candidate judge. The model receives only the reference chart and candidate chart, with the reference shown first, and is not given the requested transformation, source chart, underlying data, or generation method. It assigns a score from 0 to 10 using a rubric grounded in principles from visualization theory concerning graphical expressiveness, perceptual comparison, and information preservation. In particular, the rubric checks whether the candidate preserves the facts encoded by the reference, following the expressiveness principle that a visualization should represent all and only the intended information (Mackinlay, 1986). It therefore treats incorrect series identities, misassigned values, missing or added series, and contradictory value labels as consequential discrepancies. Changes to axis direction or range are also treated as high impact because they can alter perceived trends or magnitudes (Pandey et al., 2015). The rubric further considers changes in graphical encoding, since different encodings support different perceptual comparisons (Cleveland & McGill, 1984; Zacks & Tversky, 2013), and evaluates whether values, rankings, identities, and broader patterns remain recoverable, consistent with task-oriented accounts of visualization use (Brehmer & Munzner, 2013). In contrast, changes such as fonts, layout, or consistently applied colour palettes are treated as cosmetic unless they impair interpretation. A blank or unrelated candidate receives a score of 0, while a candidate preserving the reference's substantive information and essential encoding receives 10. Intermediate scores reflect progressively greater loss or distortion of recoverable information. The literature motivates the distinctions considered by the rubric, while the specific score ranges and decision rules are study-specific design choices. The judge also returns the detected high-impact discrepancies, remaining differences, and a short explanation. See Appendix A for the prompt and scoring rubric.

Table 3: Agreement between evaluation metrics and human judgments on CHARTFUN.

Metric	Pearson ↑	Spearman ↑	Kendall ↑	AUROC ↑
General Visual Similarity Metrics				
DreamSim	0.46	0.45	0.37	0.76
SigLIP	0.23	0.26	0.21	0.65
CLIP	0.22	0.23	0.19	0.64
SSIM	0.19	0.19	0.16	0.61
Chart-Specific Evaluation Metrics				
ChartEval (Correctness)	0.14	0.15	0.13	0.59
ChartEval (Completeness)	0.15	0.15	0.13	0.59
ChartEval (1 - Hallucination)	0.21	0.22	0.19	0.63
ChartEval (1 - Omission)	0.22	0.23	0.20	0.63
GPTScore-GPT-5	0.54	0.56	0.46	0.83
Functional-Equivalence Metrics (Ours)				
Structured Chart Comparison	0.30	0.31	0.27	0.68
Image Similarity	0.37	0.38	0.31	0.72
VLM Judge	**0.60**	**0.60**	**0.54**	**0.84**

5.2 STRUCTURED CHART COMPARISON

Our second evaluator compares two charts through structured representations. A VLM first parses each chart into the same schema, capturing its chart type, axes, categories, series, legend, values, trends, extrema, and ranks. We then compare the two representations using a deterministic set of rules. These rules detect differences such as missing series, changed values, altered trends, rank changes, axis differences, and missing metadata. The comparison is designed to focus on changes in chart meaning rather than harmless representational differences. The comparison does not penalize differences that preserve the same underlying information. For example, different numerical formatting such as 10 and 10.0 is treated as equivalent. Unordered categories may appear in a different order when their associated values are unchanged. Horizontal and vertical bar charts can also be matched by identifying which axis represents categories and which represents values.

Each discrepancy is assigned one of five severity tiers. T0 denotes information-preserving changes, T1 contextual or presentation losses, T2 magnitude distortions, T3 changes to comparisons, rankings, or extrema, and T4 contradictory facts or changes that can reverse a conclusion. These distinctions are motivated by work on representation invariance (Kindlmann & Scheidegger, 2014), graphical expressiveness (Mackinlay, 1986), visualization tasks (Brehmer & Munzner, 2013), and axis manipulation (Pandey et al., 2015). We map the five tiers to penalties of 0, 0.2, 0.4, 0.6, and 0.8. Let w_i denote the penalty of discrepancy i, and let i^* denote one with maximum severity. Then:

$$s_{\text{sem}} = \max\left(0,\, 1 - w_{i^*} - 0.25\left[1 - \prod_{i \neq i^*}(1 - w_i)\right]\right). \tag{1}$$

The most severe discrepancy therefore contributes its full penalty, while additional discrepancies contribute through a bounded term. The final score lies within $[0, 1]$. Full parsing and scoring details are provided in Appendix B.

5.3 IMAGE SIMILARITY

Our third evaluator is a lightweight model trained to predict the VLM judge scores from frozen image representations. The judge provides an integer score $J_i \in \{0, \ldots, 10\}$, which we rescale to $y_i = J_i/10$ and use as the only training target. Images are encoded with a frozen MatCha encoder (Liu et al., 2023), and the two visual representations are compared through shared bidirectional cross-attention. The resulting representation is mapped to a score by a small trained prediction head with approximately 85K trainable parameters. We train in two stages using AdamW (learning rate 3×10^{-4}, weight decay 10^{-4}, batch size 12). Stage 1 uses 4,500 source charts paired with rule-based transformations, with 250 pairs each for validation and diagnostic testing. Training stops after 19

Table 4: Class means after per-metric min–max normalization for CHARTFUN pairs.

Metric	Observed Range	Yes ↑	No ↓	Gap ↑
General Visual Similarity				
DreamSim	0.39–1.00	0.85	0.70	0.15
SigLIP	0.54–1.00	0.88	0.81	0.07
CLIP	0.34–1.00	0.86	0.80	0.06
SSIM	0.16–0.99	0.78	0.73	0.05
Chart-Specific Evaluation				
ChartEval (Correctness)	0.05–1.00	0.82	0.76	0.06
ChartEval (Completeness)	0.11–1.00	0.82	0.75	0.07
ChartEval (1-Hallucination)	0.00–1.00	0.74	0.60	0.14
ChartEval (1-Omission)	0.00–1.00	0.76	0.62	0.14
GPTScore	0.18–1.00	0.90	0.72	0.18
Functional-Equivalence Metrics (Ours)				
Structured Chart Comparison	0.00–1.00	0.49	0.26	0.23
Image Similarity	0.05–1.00	0.82	0.58	0.24
VLM Judge	0.00–1.00	0.87	0.46	**0.41**

of 100 epochs, and we select epoch 7 based on validation MAE. Stage 2 fine-tunes this checkpoint on 839 target charts paired with model-generated candidates, with 93 validation pairs. We select checkpoints using the equally weighted mean MAE across the Stage 1 and Stage 2 validation sets. Training stops after 24 epochs, with epoch 12 selected. This evaluator never saw any of the 2400 charts in CHARTFUN before testing.

We now return to Table 3 to evaluate the three functional-equivalence metrics introduced above. The VLM Judge shows the strongest agreement with human judgments, reaching Pearson and Spearman correlations of 0.60, Kendall correlation of 0.54, and AUROC of 0.84. This exceeds the strongest existing metrics, GPTScore and DreamSim, which reach Pearson correlations of 0.54 and 0.46 and AUROC of 0.83 and 0.76. The Image Similarity Evaluator reaches a Pearson correlation of 0.37 and AUROC of 0.72, while Structured Chart Comparison reaches 0.30 and 0.68, respectively. Although both trail the VLM Judge, they evaluate functional equivalence through substantially different mechanisms: explicit structured comparison and lightweight prediction from image pairs.

We min–max normalize each metric using its observed minimum and maximum across the same 2,400 pairs, with shared bounds for both classes. We then compute the mean normalized scores for majority-equivalent and non-equivalent pairs and define the gap as their difference. General visual similarity measures retain high scores for both classes, with gaps of 0.05–0.15. The VLM Judge achieves the largest gap (0.41), while Structured Chart Comparison and the Image Similarity Evaluator achieve gaps of 0.23 and 0.24, respectively. The latter provides this separation without requiring an API call at inference time.

6 CONCLUSION

We introduced CHARTFUN to study the chart editing capabilities of frontier multimodal models, asking whether a candidate preserves the information and inferences supported by a reference rather than whether the two images simply look similar. The benchmark combines 36 controlled chart transformations with 400 source charts, six generation models, and 7,200 human judgments. Human evaluation shows that functional equivalence remains challenging for current chart-generation systems, with only 60.5% of generated outputs judged equivalent overall. We further find that common visual similarity metrics align poorly with these judgments. Our visualization-guided VLM Judge provides substantially stronger agreement, while Structured Chart Comparison and Image Similarity explore complementary structured and lightweight alternatives. Together, these results suggest that chart evaluation should explicitly account for the information a visualization communicates, not only its visual resemblance to a reference.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data sets (the candidate charts for CHARTFUN) and to implement methods (the VLM Judge and Image Similarity). We have not used generative AI for any other required-disclosure tasks. Additionally, we used AI to identify relevant literature, create or modify scientific figures, and edit the research paper to improve readability. The authors reviewed all AI-assisted work. Specifically, the generated charts and metric outputs were validated via our human evaluation study, and initial figures were refined by hand. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Matthew Brehmer and Tamara Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, December 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.124. URL https://doi.org/10.1109/TVCG.2013.124.

William S. Cleveland and Robert McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984. ISSN 01621459, 1537274X. URL http://www.jstor.org/stable/2288400.

Ondřej Dušek and Zdeněk Kasner. Evaluating semantic accuracy of data-to-text generation with natural language inference. In Brian Davis, Yvette Graham, John Kelleher, and Yaji Sripada (eds.), *Proceedings of the 13th International Conference on Natural Language Generation*, pp. 131–137, Dublin, Ireland, December 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.inlg-1.19. URL https://aclanthology.org/2020.inlg-1.19/.

Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In *Advances in Neural Information Processing Systems*, volume 36, pp. 50742–50768, 2023.

Google DeepMind. Gemini 3.6 Flash model card. Google DeepMind Model Card, July 2026.

Kanika Goswami, Puneet Mathur, Ryan A. Rossi, Franck Dernoncourt, Vivek Gupta, and Dinesh Manocha. ChartEval: LLM-driven chart generation evaluation using scene graph parsing. In Xuebo Liu and Ayu Purwarianti (eds.), *Proceedings of The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 86–93, Mumbai, India, December 2025. Association for Computational Linguistics. ISBN 979-8-89176-301-2. doi: 10.18653/v1/2025.ijcnlp-demo.10. URL https://aclanthology.org/2025.ijcnlp-demo.10/.

Gordon L. Kindlmann and Carlos Eduardo Scheidegger. An algebraic process for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 20:2181–2190, 2014. URL https://api.semanticscholar.org/CorpusID:10535011.

Jill H. Larkin and Herbert A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11(1):65–100, 1987. ISSN 0364-0213. doi: https://doi.org/10.1016/S0364-0213(87)80026-5. URL https://www.sciencedirect.com/science/article/pii/S0364021387800265.

Claire Lauer and Shaun O'Brien. The deceptive potential of common design tactics used in data visualizations. In *Proceedings of the 38th ACM International Conference on Design of Communication*, SIGDOC '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375252. doi: 10.1145/3380851.3416762. URL https://doi.org/10.1145/3380851.3416762.

Shawn Li, Ryan Rossi, Sungchul Kim, Sunav Choudhary, Franck Dernoncourt, Puneet Mathur, Zhengzhong Tu, and Yue Zhao. Charts are not images: On the challenges of scientific chart editing, 2025. URL https://arxiv.org/abs/2512.00752.

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 (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12756–12770, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.714. URL https://aclanthology.org/2023.acl-long.714/.

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?, 2022. URL https://arxiv.org/abs/2204.09548.

Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, April 1986. ISSN 0730-0301. doi: 10.1145/22949.22950. URL https://doi.org/10.1145/22949.22950.

Andrew McNutt, Gordon Kindlmann, and Michael Correll. Surfacing visualization mirages. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, pp. 1–16, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376420. URL https://doi.org/10.1145/3313831.3376420.

Nitesh Methani, Pritha Ganguly, Mitesh M. Khapra, and Pratyush Kumar. Data interpretation over plots. *CoRR*, abs/1909.00997, 2019. URL http://arxiv.org/abs/1909.00997.

OpenAI. GPT-5.5 system card. OpenAI System Card, April 2026a.

OpenAI. GPT-Image-2: Model documentation. OpenAI API Documentation, 2026b. Accessed September 26, 2026.

Anshul Vikram Pandey, Katharina Rall, Margaret L. Satterthwaite, Oded Nov, and Enrico Bertini. How deceptive are deceptive visualizations? an empirical analysis of common distortion techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 1469–1478, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450331456. doi: 10.1145/2702123.2702608. URL https://doi.org/10.1145/2702123.2702608.

Steven Pinker. A theory of graph comprehension. In Roy O. Freedle (ed.), *Artificial Intelligence and the Future of Testing*, pp. 73–126. Lawrence Erlbaum Associates, Hillsdale, NJ, 1990.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Naina Raisinghani. Introducing Nano Banana Pro (Gemini 3 Pro Image). Google Blog, November 2025.

Raj Ratwani and J. Trafton. Shedding light on the graph schema: Perceptual features versus invariant structure. *Psychonomic Bulletin Review*, 15:757–762, 09 2008. doi: 10.3758/PBR.15.4.757.

Priti Shah and Eric Freedman. Bar and line graph comprehension: An interaction of top-down and bottom-up processes. *Topics in Cognitive Science*, 3:560–578, 07 2011. doi: 10.1111/j.1756-8765.2009.01066.x.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004a. doi: 10.1109/TIP.2003.819861.

Zhou Wang, Alan Bovik, H.R. Sheikh, and Eero Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 05 2004b. doi: 10.1109/TIP.2003.819861.

Chengyue Wu, Zhixuan Liang, Yixiao Ge, Qiushan Guo, Zeyu Lu, Jiahao Wang, Ying Shan, and Ping Luo. Plot2Code: A comprehensive benchmark for evaluating multi-modal large language models in code generation from scientific plots. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3006–3028, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.164. URL https://aclanthology.org/2025.findings-naacl.164/.

Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Peng Ye, Min Dou, Botian Shi, Junchi Yan, and Yu Qiao. Chartx chartvlm: A versatile benchmark and foundation model for complicated chart reasoning, 2025. URL https://arxiv.org/abs/2402.12185.

Pengyu Yan, Mahesh Bhosale, Jay Lal, Bikhyat Adhikari, and David Doermann. Chartreformer: Natural language-driven chart image editing, 2024. URL https://arxiv.org/abs/2403.00209.

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, Gongye Liu, Xiaomei Nie, Deng Cai, and Yujiu Yang. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation, 2025. URL https://arxiv.org/abs/2406.09961.

Zhiyu Yang, Zihan Zhou, Shuo Wang, Xin Cong, Xu Han, Yukun Yan, Zhenghao Liu, Zhixing Tan, Pengyuan Liu, Dong Yu, Zhiyuan Liu, Xiaodong Shi, and Maosong Sun. Matplotagent: Method and evaluation for llm-based agentic scientific data visualization, 2024. URL https://arxiv.org/abs/2402.11453.

Jeff Zacks and Barbara Tversky. Bars and lines: A study of graphic communication. *Memory Cognition*, 27:1073–1079, 03 2013. doi: 10.3758/BF03201236.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986, October 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CoRR*, abs/1801.03924, 2018. URL http://arxiv.org/abs/1801.03924.

Xuanle Zhao, Xuexin Liu, Haoyue Yang, Xianzhen Luo, Fanhu Zeng, Jianling Li, Qi Shi, and Chi Chen. Chartedit: How far are mllms from automating chart analysis? evaluating mllms' capability via chart editing, 2025. URL https://arxiv.org/abs/2505.11935.

A VLM JUDGE PROMPT

We reproduce below the complete prompt used by the VLM judge. The reference image is provided first and the candidate image second.

System message.

```
Given a reference chart and a candidate chart, evaluate how well the
candidate preserves the reference chart's semantic content, supported
inferences, and essential visual encoding. Assign an integer score from
0 to 10.

Judge only the visible images. Do not infer the intended edit, generation
method, hidden source data, or expected severity.

Functional equivalence requires preservation of trends, approximate
magnitudes, rankings, series identities, and conclusions. Differences
in fonts, layout, gridlines, or consistently applied color palettes are
cosmetic unless they impair interpretation.

First check whether the candidate is blank, unrelated, or contains no
recoverable reference information. If so, assign 0, report this in
Stage 1, and mark Stage 2 as "skipped due to Stage 1 failure". This
zero-score rule takes precedence over all remaining score ranges.
Otherwise, continue with the checks below.

Stage 1: High-impact discrepancies

Check the conditions below before evaluating smaller differences. A
detected discrepancy restricts the score to the specified range. If
multiple conditions apply, use the lowest applicable score ceiling.
Within a range, assign lower scores to systematic or conclusion-changing
errors and higher scores to isolated or recoverable errors.

Treat legend-swatch colors and plotted-series colors as independent
observations. Establish legend-to-series correspondence using non-color
evidence, such as direct labels, marker style, line style, or bar-group
position. Never use legend order or pie-wedge order as evidence of
series identity. A consistent palette change is not an error. Do not
apply a legend-related restriction when correspondence is uncertain.

Series identity and legend binding: A legend label denotes the wrong
plotted series, its swatch conflicts with the corresponding mark, or
series identities are clearly exchanged. Score: 1-2.

Data assignment: Values are systematically assigned to the wrong series,
categories, or time points. Score: 1-2.

Axis direction: A numeric, temporal, or ordinal axis is reversed in a way
that changes the inferred trend. Score: 1-2.

Axis range: A materially altered minimum or maximum exaggerates or
suppresses differences. Score: 1-3.

Scale type: A linear-logarithmic change affects magnitude judgments.
Different tick spacing on the same scale is excluded. Score: 1-3.

Bar-line encoding: A conversion between bars and a connected line when
all categories, values, and series identities are preserved. Score: 5.

Scatter or pie encoding: A conversion to or from scatter or pie changes
the supported comparisons or basis of interpretation. Score: 2-4.

Stacking: Grouped series become stacked, or stacked series become
↪ grouped.
Score: 2-4.

Series inventory: A series is added or omitted. Added series score 1-3.
Omitted series score 2-4 according to importance. Score: 1-4.

Dual axes: A secondary value axis is added, removed, or reassigned in a
```

```
way that changes cross-series comparison. Score: 4-6.

Value labels: Candidate value labels contradict the corresponding marks
or reference values. Inspect two or three whenever candidate labels are
present, including when the reference has none. An isolated error may
score up to 3. Score: 1-3.

Title meaning: A displayed candidate title is semantically unrelated to
the reference title. Missing titles are handled in Stage 2. Score: 3-5.

Reordering nominal categories is not a high-impact discrepancy unless
their order conveys meaning. A missing legend alone does not constitute
an omitted series. Isolated quantitative deviations are evaluated in
Stage 2.

Stage 2: Standard comparison and scoring

If no Stage 1 discrepancy is detected, compare:

chart-title presence and meaning
x- and y-axis titles
tick-label sets and ordering
legend presence and entry set
numbers of plotted series and categories
two or three corresponding data values
remaining chart-type or encoding differences

When legend removal is the principal difference, score it according to
recoverability: use 9 when the legend is redundant and series identity
remains evident, 6 when identities remain inferable only with effort,
and 5 when multi-series identities are unrecoverable but the plotted
patterns remain.

Use the following score anchors:

10: All substantive and contextual information matches. Cosmetic
differences are allowed.

9: A negligible difference, including omission of a redundant legend
when series identity remains evident, does not affect interpretation.

8: A minor presentation or contextual difference increases
interpretation effort but preserves conclusions.

7: One of the chart title, x-axis title, or y-axis title is missing.
Data, ticks, and remaining identifying information remain intact.

6: At least two chart or axis titles are missing, or a missing legend
makes series identities recoverable only with effort.

5: Identifying metadata or multi-series identities are unrecoverable,
but the plotted values and main pattern remain preserved.

4: One non-critical data, tick, or range element is incorrect, producing
a localized error.

3: Multiple metadata errors and at least one quantitative error produce
incorrect details, although the chart remains recognizably related.

2: A high-impact discrepancy changes a central conclusion, but some
relevant content remains.

1: Only minimal relevant information remains. The reference's main
answers and conclusions are largely unrecoverable.

0: The candidate is blank, unrelated, or contains no recoverable
```

```
reference information.

Output format

Return exactly four single-line entries:

Stage 1: up to six semicolon-separated high-impact discrepancies, or
"all high-impact checks passed"

Stage 2: up to six semicolon-separated differences, "all match," or
"skipped due to Stage 1 failure"

Reasoning: one sentence of at most 30 words

{"score": N}

The final line must contain only the JSON object, where N is an integer
from 0 to 10.
```

User message.

```
The reference chart is shown first and the candidate chart second.
Evaluate the candidate using the rubric above. Judge only visible
evidence. Apply the zero-score rule first. Otherwise, any Stage 1
discrepancy restricts the score to its specified range.
```

B STRUCTURED CHART PARSER PROMPT

We reproduce below the prompt used to obtain the structured chart representation. The scoring stage operates on the resulting structured representations and does not use an additional VLM prompt.

System message.

```
Given one chart image, extract a structured description. Return exactly
one valid JSON object conforming to the schema below, without prose or
code fences. Use null when information is absent, unreadable,
inapplicable, or ambiguous. Do not invent content.

{
  "title": <string or null>,
  "subtitle": <string or null>,
  "chart_type": <"line" | "scatter" | "bar_v" | "bar_h" | "pie" |
                 "other" | null>,

  "x_axis": {
    "title": <string or null>,
    "tick_labels": [<string>, ...],
    "is_categorical": <bool or null>,
    "scale_type": <"linear" | "log" | "categorical" | "date" | null>,
    "baseline": <number or null>,
    "range_max": <number or null>,
    "direction": <"normal" | "inverted" | null>,
    "tick_intervals_uniform": <bool or null>
  },

  "y_axis": {
    "title": <string or null>,
    "tick_labels": [<string>, ...],
    "is_categorical": <bool or null>,
    "scale_type": <"linear" | "log" | "categorical" | "date" | null>,
    "baseline": <number or null>,
    "range_max": <number or null>,
    "direction": <"normal" | "inverted" | null>,
    "tick_intervals_uniform": <bool or null>
```

```
  },

  "legend": {
    "present": <bool>,
    "position": <"right" | "bottom" | "top" | "left" | "inside" |
    ↪  "none">,
    "entries": [
      {
        "label": <string or null>,
        "color": <string or null>
      }
    ]
  },

  "series_count": <int>,
  "series": [
    {
      "label": <string or null>,
      "label_source": <"direct" | "legend_noncolor" |
                       "legend_color" | null>,
      "color": <string or null>,
      "marker_shape": <"circle" | "square" | "triangle" | "diamond" |
                       "line_only" | "bar" | "none" | null>,
      "line_style": <"solid" | "dashed" | "dotted" | "none" | null>,
      "approx_n_points": <int or null>,
      "magnitude_share": <number or null>,

      "value_summary": {
        "trend_direction": <"increasing" | "decreasing" | "flat" |
                            "mixed" | null>,
        "argmax_x_label": <string or null>,
        "argmin_x_label": <string or null>,
        "first_x_label": <string or null>,
        "last_x_label": <string or null>,
        "first_y_value": <number or null>,
        "last_y_value": <number or null>,
        "rank_at_first_x": <int or null>,
        "rank_at_last_x": <int or null>
      },

      "value_label_texts": [<string>, ...] | null
    }
  ],

  "chart_mode": {
    "stack_mode": <"none" | "stacked" | "grouped" | "percent_stacked">,
    "is_dual_axis": <bool>,
    "uniform_bar_widths": <bool or null>
  },

  "gridlines": {
    "horizontal": <bool>,
    "vertical": <bool>
  },

  "annotations": {
    "data_labels": <bool>,
    "text_annotations_count": <int>,
    "highlights_count": <int>
  },

  "self_check": {
    "legend_labels_match_series_labels": <bool or null>,
    "legend_binding_consistent": <bool or null>,
    "data_label_count_matches_points": <bool or null>,
    "n_series_matches_legend_entries": <bool or null>
```

```
  }
}

Apply these rules:

1. Chart type and axes: Use line for connected lines, with or without
point markers, and scatter for unconnected points. Use other for a
visible unsupported chart type and null when the type cannot be
determined.

For charts with axes, record x-axis ticks left-to-right and y-axis ticks
bottom-to-top; record horizontal-bar category labels top-to-bottom. Use
an empty list when no ticks are visible. For categorical axes, set
baseline, range_max, and tick_intervals_uniform to null. For numeric
axes, use the lowest and highest visible ticks as baseline and range_max;
do not assume zero. Set direction according to whether values increase
in the conventional direction.

Complete axis objects are required. For both axes of a pie chart, use:

{
  "title": null,
  "tick_labels": [],
  "is_categorical": null,
  "scale_type": null,
  "baseline": null,
  "range_max": null,
  "direction": null,
  "tick_intervals_uniform": null
}

2. Series: Count series from plotted marks rather than legend entries.

* line: each distinct plotted line is one series.
* scatter: each visually distinct or legend-defined point group is one
  series, not each point.
* bar_v: bars with the same repeated role across x-axis categories form
  one series.
* bar_h: bars with the same repeated role across y-axis categories form
  one series.
* pie: each visible wedge is one series.

In grouped or stacked bars, each repeated subgroup or stack component is
a separate series.

3. Legend association: Record legend-swatch and plotted-series colors
independently, using names specific enough to distinguish visible shades.
Set label_source to:

* direct when the label is printed with the plotted mark;
* legend_noncolor when the legend association is supported by marker or
  line style, bar position, plotted ordering cues, or elimination;
* legend_color when a unique color match is the only evidence;
* null when no reliable association exists.

Do not use legend order alone to establish series identity. Color-only
association is allowed for all chart types. When correspondence relies
only on color, set legend_binding_consistent to null. Set it to false if
an independently established legend-series pair has different colors,
and to true only when all relevant pairs are independently established
and match. Never alter a plotted color to agree with the legend.

4. Values: For line, scatter, and bar charts, estimate

(series maximum - axis baseline) / (axis maximum - axis baseline)
```

```
as magnitude_share, or use null when unreliable. Use the numeric value
axis: normally the y-axis, but the x-axis for horizontal bars and
horizontally oriented scatter plots. For stacked bars, use segment
thickness rather than its cumulative endpoint.

Approximate value_summary values are acceptable; preserve trends,
ordering, and extrema. Use 1-based ranks and report ranks or extrema only
when visually unambiguous. For horizontal bars, fields ending in
_x_label refer to categorical y-axis labels. For scatter plots, use
trend_direction for the overall x-y relationship when visually clear.

For each pie wedge, set approx_n_points to 1, marker_shape and line_style
to null, and every value_summary field to null. Estimate magnitude_share
from wedge angle independently of printed values; wedge shares should
sum approximately to 1. Wedge order does not establish category identity.

5. Labels and annotations: If numeric or percentage labels appear beside
marks, set annotations.data_labels to true and transcribe them exactly
in plot order. Store printed labels separately from geometrically
estimated values. Use "?" for an unreadable label and null when a series
has no visible value labels.

Labels attached directly to a plotted series or pie wedge may populate
series.label; axis tick labels must not. Category labels are not data
labels. Count other free-text annotations separately from titles and
legends; count arrows, boxes, and similar emphasis marks as highlights.

6. Chart mode and checks: Use the observed bar arrangement to set
stack_mode; use "none" when stacking does not apply. Set is_dual_axis to
true only when a secondary value axis is visible. Set
uniform_bar_widths to null for non-bar charts. Set both gridline fields
to false for pie charts.

Compute self-checks only when the necessary evidence is available;
otherwise use null.
```

User message.

```
Parse the provided chart image.
```

C Implementation Details

Benchmark and human study. The benchmark has 400 reference charts from the PlotQA test split, 80 per chart type (horizontal bar, vertical bar, line, line with markers, pie). Each reference applies 2–5 edits, with 100 references per edit count. Six generators (GPT-5.5, GPT-Image-2, Gemini-3-Pro-Image, Gemini-3.6-Flash, Qwen3.8-27B, InternVL3.5-38B) each produce one candidate per reference, giving 2,400 pairs. Each pair was rated by three of 93 Prolific participants (7,200 votes), who answered whether the two charts convey the same information. The reference was shown on the left or right at random. The target is the majority vote, which gives 1,451 equivalent and 949 non-equivalent pairs.

VLM Judge. The Judge runs GPT-5. It receives both charts, reference first, as RGB PNG. Output is capped at 6,500 completion tokens. The response must end with an integer score in $[0, 10]$, which we divide by 10.

Judge-distilled student. The student is trained on Judge scores only, with target $y = s/10$. A frozen MatCha-base encoder (Liu et al., 2023) encodes each chart at its native aspect ratio (up to 2,048 patches). A small head projects the tokens to 64 dimensions and applies one bidirectional cross-attention block (4 heads). It then maps the pooled pair feature $[r; c; c - r; |c - r|]$ to a sigmoid score. Training has two stages on PlotQA training charts, none of which appear in the benchmark (Table 5). Stage 1 uses 5,000 rule-edited pairs (4,500 training, 250 validation, 250 diagnostic test).

Stage 2 starts from the Stage 1 checkpoint and uses 932 generator-output pairs (839 training, 93 validation).

Hyperparameter	Value
Optimizer	AdamW, lr 3×10^{-4}, weight decay 10^{-4}
Schedule	Cosine decay to 3×10^{-5}
Batch size	12
Epochs	Up to 100; early stopping with patience 12
Loss	Smooth-L1 ($\beta = 0.1$)
Gradient clipping / dropout	1.0 / 0.1
Selected epoch	Stage 1: 7 of 19; Stage 2: 12 of 24
Hardware	$1\times$ A100 (40 GB), about 2 minutes in total

Table 5: Student training hyperparameters.

Baselines.

- DreamSim (ensemble) (Fu et al., 2023), CLIP (ViT-L/14) (Radford et al., 2021) and SigLIP (so400m-patch14-384) (Zhai et al., 2023) score a pair by the cosine similarity of their image embeddings.

Evaluation. Every metric is correlated with the binary majority vote over all 2,400 pairs using Pearson (point-biserial), Spearman, Kendall τ_b and AUROC. Scores are used exactly as each metric outputs them, with no fitting, calibration or thresholding.