ChartEval: LLM-Driven Chart Generation Evaluation Using Scene Graph Parsing

Kanika Goswami[1], Puneet Mathur[2], Ryan Rossi[2], Franck Dernoncourt[2], Vivek Gupta[3], Dinesh Manocha[4]

[1]IGDTUW, India

[2]Adobe Research, San Jose, USA

[3]Arizona State University, Tempe, USA

[4]University of Maryland, College Park, USA

Demo Video: https://youtu.be/HcPuJaVO04s **Demo Link**: chartEval.ai

Abstract

Accurate assessment of generated chart quality is crucial for automated document creation and editing across diverse applications like finance, medicine, policy making, and education. Current evaluation approaches suffer from significant limitations: human evaluation is costly and difficult to scale, pixel-based metrics ignore data accuracy, while data-centric measures overlook design quality. Recent multimodal LLM evaluators show promise but exhibit concerning inconsistencies due to prompt sensitivity and subjective biases. Existing metrics fail to evaluate chart quality holistically across visual similarity, semantic alignment, and data fidelity, often producing misleading scores that unfairly penalize good charts while rewarding bad ones. We introduce **ChartEval**, a novel chart evaluation system that compares generated chart images with ground truth by leveraging scene graph parsing to decompose chart images into hierarchical scene graphs of chart objects, attributes, and relations. Subsequently, it applies graph-based similarity measures to compare candidate chart scene graphs against reference scene graphs for measuring chart quality. We demonstrate that our evaluation approach achieves significantly stronger correlation with human judgments compared to existing metrics like GPT-Score, SSIM, and SCRM using a comprehensive benchmark of 4K chart images paired with generation intents and human quality ratings. We demonstrate the utility of the ChartEval system as a reliable automatic chart quality metric on diverse tasks including *language-guided chart editing, chart reconstruction, and text-to-chart synthesis* using both open-source and API-based LLMs. **Demo Website & Video**: chartEval.ai

1 Introduction

Effective data visualization transforms vast amounts of information into actionable insights, playing a critical role across professional domains including financial reporting, scientific publishing, policy analysis, and clinical documentation. However, creating high-quality charts requires substantial technical expertise, driving growing demand for automated chart generation systems. While chart question-answering and captioning have been extensively studied, text-to-chart generation and chart editing have recently gained significant attention as Large Language Models (LLMs) enable users to create visualizations through natural language.

Despite these advances, evaluating chart quality presents significant challenges that existing metrics fail to address comprehensively. Human evaluation, while thorough, is costly and impractical for scaling across large datasets with diverse visualization types. Existing automated evaluation methods, though scalable, suffer from fundamental limitations that compromise their reliability. Data-centric approaches such as SCRM (Xia et al., 2023) extract underlying data tables from charts, but focus exclusively on data accuracy while ignoring visual design quality like mismatched color schemes, misleading labels, or cluttered layouts. Pixel-based metrics like SSIM perform direct image comparisons at the pixel level (Yan et al., 2024), but unfairly penalize semantically equivalent charts that exhibit minor visual differences due to different rendering libraries or styling choices. LLM-as-a-judge methods leverage multimodal LLM prompting to assess generated visualizations (Shi et al., 2025; Xia et al., 2024), offering better scalability but suffering from inconsistent outputs due to prompt sensitivity and subjective biases. These limitations result in evaluation systems that frequently mischaracterize chart quality, incorrectly penalizing well-designed charts while rewarding ones with poor visual communication.

We propose **ChartEval** (Fig.1) - a novel chart evaluation system that views chart images as visual scene graphs. In this representation, visual objects such as data marks and legends form nodes,

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, pages 86–93

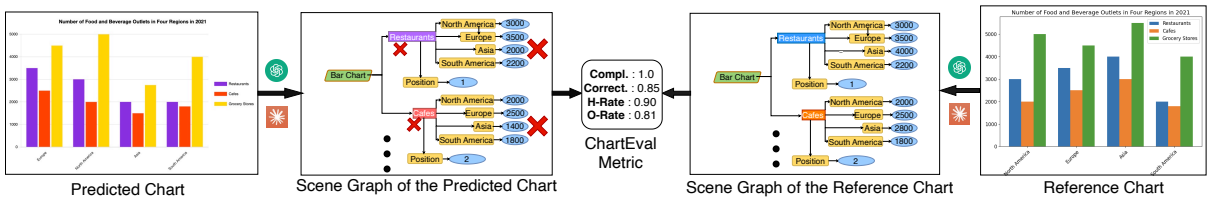

Figure 1: **ChartEval** evaluates chart quality by (1) decomposing predicted and reference charts into visual scene graphs via ChartSceneParse prompting using multimodal LLMs; (2) applying graph-based similarity measures: Graph-BERTScore for semantic correctness and completeness, Hallucination Rate for spurious content, and Omission Rate for missing information. Red crosses (×) highlight values in scene graph of candidate chart that do not match the ground truth chart.

with each object defined by attributes like colors, sizes, and positions, while edges capture relations such as spatial arrangements and data mappings between objects. Given a candidate chart image and its ground truth reference, `ChartEval` decomposes both charts into structured representations using standardized grammar formats (e.g., Vega JSON specifications) that captures overall chart semantics. To enable this decomposition, we introduce `ChartSceneParse`, a novel prompting technique that leverages Chain-of-Thought reasoning (Wei et al., 2022) to systematically extract scene graphs from chart images using multimodal LLMs. We compute graph similarity between the extracted scene graphs of predicted and ground truth charts using four complementary metrics to judge for semantic correctness, completeness, hallucination of spurious content, and omission of critical components. Users can utilize `ChartEval` to evaluate charts generated via multiple methods, including but not limited to SOTA open-source and API-based multimodal LLMs, considering their semantic similarity, visual alignment, and data fidelity.

Lastly, we propose a new benchmark - `ChartGen` comprising of 4K diverse reference chart images paired with their generation intents and human quality ratings across three diverse tasks: language-guided chart editing, chart reconstruction, and text-to-chart synthesis. This benchmark was assembled by integrating four open-source datasets - ChartCraft, ChartMimic, ChartX, and Text2Chart31. We validate the performance of `ChartEval` on `ChartGen` benchmark to demonstrate that `ChartEval` achieves significantly stronger correlation with human judgments compared to existing metrics across most scenarios, confirming its effectiveness as a reliable chart evaluation tool. Results show that participants find the `ChartEval` metric to be an accurate and reliable metric for fact checking generated charts.

Our **main technical contributions** are:

- **ChartSceneParse** prompting technique that leverages Chain-of-Thought reasoning to systematically extract hierarchical scene graphs from chart images using multimodal LLMs and Vega JSON grammar.
- **ChartEval** system that compares generated charts with ground truth by decomposing both into scene graphs and applying graph-based similarity measures (Graph-BERTScore, Hallucination Rate, Omission Rate) for comprehensive quality assessment.
- **ChartGen benchmark** of 4K diverse chart images with human quality ratings across chart editing, reconstruction, and text-to-chart synthesis tasks, achieving significantly stronger correlation with human judgments than existing metrics.

Our **main system-level contributions** are:
(1) Interpretability: `ChartEval` promotes interpretable chart evaluation by providing granular descriptions of visual, semantic, and data hallucinations/omissions through `ChartSceneParse`.
(2) Explainability: `ChartEval` provides logical rationales alongside metric scores to clarify the reasoning behind identified hallucinations and omissions. By transforming charts into hierarchical entity representations, it transcends simple visual similarity metrics and enables direct attribution to chart metadata.
(3) Reliability: `ChartEval` assists professionals across business, education, and finance domains by reducing time spent fact-checking generated charts, allowing users to focus on more productive tasks while enhancing overall evaluation reliability.

2 `ChartEval` - Target Audience

ChartEval's scene graph representation unlocks powerful reference-free applications by transform-

ing charts into structured, analyzable formats: (1) **Automated Quality Control**: Analyze scene graph structure to detect missing legends, unlabeled axes, or inconsistent data encodings in document editing workflows. (2) **Enterprise Style Compliance**: Define corporate chart standards as scene graph templates to ensure all generated charts follow consistent color schemes, font choices, and layout patterns across reports. (3) **Cross-Chart Consistency**: Verify that multiple visualizations within documents use compatible scales, similar encoding principles, and coherent design languages. (4) **Data Integrity Validation**: Compare extracted scene graph data points against source datasets to automatically flag discrepancies, incorrect calculations, or missing information without requiring reference charts. (5) **Collaborative Quality Standards**: Enable teams to maintain quality standards by detecting when charts violate readability principles, accessibility guidelines, or domain-specific conventions. (6) **Template-Based Generation**: Allow users to define desired chart patterns as scene graphs, then automatically evaluate whether generated visualizations match these structural requirements. This structured representation transforms chart evaluation from purely comparative assessment to comprehensive quality analysis, enabling automated editorial assistance, style enforcement, and accuracy verification in production document workflows where reference charts don't exist.

3 `ChartEval` - System Architecture

`ChartEval` (Fig.1) evaluates the representational fidelity of a candidate chart against a ground truth chart in two stages. First, it employs `ChartSceneParse` prompting to decompose the chart images into a structured grammar representations (eg. Vega Json) with a standardized taxonomy to construct hierarchical scene graphs. Second, we compare the extracted chart scene graphs using three complementary evaluation metrics: Graph-BERTScore for semantic similarity, Hallucination Rate for spurious content detection, and Omission Rate for missing information assessment.

3.1 Chart Scene Graph Parsing

`ChartEval` decomposes chart images into structured scene graphs in three steps:

(1) Chart Structure parsing describes data visualization designs into a declarative JSON specification language by leveraging the Vega visualization grammar[1] (Satyanarayan et al., 2016). Vega grammar represents charts as hierarchical compositions of primitive graphical properties such as view dimensions, data definitions, map scales, axes, legends, marks (like lines, points, bars), and symbols that encode the underlying data, neatly organized into nested groups with explicit coordinate systems and data bindings. By structuring the extraction process around these core Vega primitives, `ChartSceneParse` systematically converts visual chart elements into their corresponding declarative representations, enabling precise reconstruction and analysis of the original visualizations.

We employ `ChartSceneParse` prompting, an LLM-based Chain-of-Thought reasoning (Wei et al., 2022) technique to systematically extract chart elements as Vega structural primitives: (i) mark types (line, bar, point) and chart layout, (ii) titles and axis labels with exact transcriptions, (iii) axis components (domains, ticks, and grid lines), visual properties (stroke, fill, opacity), and (iv) data points with both visual coordinates and semantic values. We provide prompt instructions to use the identified axis domains and tick positions as spatial anchors for accurate coordinate mappings of data points, pairing pixel positions with actual data values. The extraction follows Vega's hierarchy—first identifying the root frame, then cataloging nested components (axes, marks, titles) with their functional roles. Each extracted element is mapped to Vega's declarative format where visual properties become explicit JSON attributes and spatial relationships are encoded through coordinate transformations. For charts with incomplete information, the system infers reasonable scales while flagging approximations. The LLM is prompted to provide exact textual transcriptions of all visible labels and numerical values to mitigate any hallucinations. Few-shot examples guide the LLM to enforce Vega grammar compliance such that it preserves the proper z-ordering and coordinate system relationships between chart annotations to maintain visual parity with the source image.

(2) Self-Reflection Prompting: LLM-based parsing may be prone to hallucinations which need to be mitigated to avoid spurious results. Hence, we utilize Altair API [2] to convert the intermediate Vega specification back into a chart image and its corresponding data table. The intermediate chart

[1] https://vega.github.io/vega/docs/
[2] https://altair-viz.github.io/

image, its data table, and the reference chart image are sent to GPT-5 to illicit a match score (0-10) and a comparative feedback via Reflexion (Shinn et al., 2023) to correct the generated Scene Graph in the next iteration. We continue this iterative process until match score > 8 or maximum of 3 rounds.

(3) Scene Graph Construction – The Vega grammar JSON is converted into a directed graph $G = (V, E)$ where vertices represent chart components and edges encode their relationships. **Node Creation**: The algorithm generates typed vertices $v_i \in V$ for each functional component—title nodes (v_{title}), chart-type nodes (v_{type}), axis nodes ($v_{\text{x-axis}}, v_{\text{y-axis}}$), and data-point nodes (v_{data_i}). Each node stores attributes extracted from corresponding Vega elements: data nodes contain both visual coordinates $(x_{\text{pixel}}, y_{\text{pixel}})$ and semantic values $(x_{\text{data}}, y_{\text{data}})$, while axis nodes store domain information and labels. **Edge Formation**: Directed edges $e_{ij} \in E$ establish semantic relationships — data-to-axis edges $(v_{\text{data}_i}, v_{\text{x-axis}})$ and $(v_{\text{data}_i}, v_{\text{y-axis}})$ encode which axes govern each data point's positioning, while sequential edges $(v_{\text{data}_i}, v_{\text{data}_{i+1}})$ connect consecutive points to preserve spatial ordering. **Graph Attributes**: Node and edge attributes capture multi-level abstractions — visual properties (colors, styling), semantic content (trends, statistical summaries), and structural metadata (chart type, dimensions). This graph representation G_{chart} standardizes heterogeneous visualizations into a unified format enabling systematic structural comparison while preserving both geometric layout and data semantics.

3.2 Graph-based Scoring

After obtaining scene graphs $G_{\text{gt}} = (V_{\text{gt}}, E_{\text{gt}})$ and $G_{\text{pred}} = (V_{\text{pred}}, E_{\text{pred}})$ from ground truth and predicted charts respectively, we employ Graph-BERTScore (G-BS) (Saha et al., 2021), a semantic-level metric which extends the BERTScore (Zhang et al., 2019) for graph matching. Each edge in the graph is considered as a sentence and BERTScore is used to calculate the score between a pair of predicted and ground-truth edges. Both graphs are decomposed into semantic statements $S = \{s_1, s_2, \ldots, s_k\}$ encoding chart components (e.g., "X-axis represents: Year", "Data trend: increasing from 2010 to 2020"). We use pre-trained BERT (Devlin et al., 2019) contextual embeddings $\mathbf{e}_i = \text{BERT}(s_i)$ and compute pairwise cosine similarities $M_{ij} = \frac{\mathbf{e}_i^{\text{gt}} \cdot \mathbf{e}_j^{\text{pred}}}{||\mathbf{e}_i^{\text{gt}}|| \cdot ||\mathbf{e}_j^{\text{pred}}||}$ between all statement pairs. Following recent work in graph evaluation (Ghanem and Cruz, 2024), we calculate:

Statistic	ChartCraft	ChartMimic	ChartX	Text2Chart31
# charts (test)	5550	500/500	6000	1423
# human rated	1000	1000	1000	1000
Task	Editing	Editing/Reconstruction	Reconstruction	Desc-to-Chart
Input Format	Image + NL	Image + NL	Image	Text
Eval Metric	SSIM	GPT-Score	SCRM & GPT-score	CodeBLEU
Source	Synthetic	Human	Synthetic	Synthetic

Table 1: Comparative data statistics

(i) Correctness: Average of the maximum similarity scores between each predicted statement in S_{pred} and all ground truth statements S_{gt}: $P = \frac{1}{|S_{\text{pred}}|} \sum_{j=1}^{|S_{\text{pred}}|} \max_i M_{ij}$, analogous to precision.

(ii) Completeness: Average of the maximum similarity scores between each ground truth statement in S_{gt} and all predicted statements S_{pred}: $R = \frac{1}{|S_{\text{gt}}|} \sum_{i=1}^{|S_{\text{gt}}|} \max_j M_{ij}$, analogous to recall.

We perform direct scene graph comparison via:

(iii) Hallucination Rate (Ghanem and Cruz, 2024) quantifies spurious information in predictions. Hallucinations are defined as the presence of an entity or relation in predicted graph that is not present in the gold graph. We extract structured element sets $\mathcal{E}_{\text{gt}}$ and $\mathcal{E}_{\text{pred}}$ from the ground truth and predicted graphs, encompassing data points (x_i, y_i), axis labels, chart metadata, and visual properties. Mathematically, $H_{\text{rate}} = \frac{|\mathcal{E}_{\text{pred}} - \mathcal{E}_{\text{gt}}|}{|\mathcal{E}_{\text{pred}}|}$. We employ fuzzy matching with ϵ-tolerance for numerical values to account for minor coordinate differences while preserving semantic accuracy.

(iv) Omission Rate (Ghanem and Cruz, 2024) accounts for critical missing elements that compromise chart completeness, such as absent data points, unlabeled axes, or missing titles. Omissions are defined as missing entities or relations in the predicted graph that are present in the gold graph. Mathematically, $O_{\text{rate}} = \frac{|\mathcal{E}_{\text{gt}} - \mathcal{E}_{\text{pred}}|}{|\mathcal{E}_{\text{gt}}|}$.

4 System Demonstration

Figure 2 shows the `ChartEval` demo app (chartEval.ai) which has been created using Gradio, and can use OpenAI GPT-4o or Claude Sonnet-3.7 for chart evaluation. The interface includes a panel to upload pairs of predicted/reference chart images and an alternative option to select from pre-uploaded examples. The user can choose which LLM to use for the evaluation from the UI and also reset the interface for new evaluations. `ChartEval` shows an evaluation report with scores for completeness, correction, Hallucination Rate, and Omission Rate, along with an explanation on

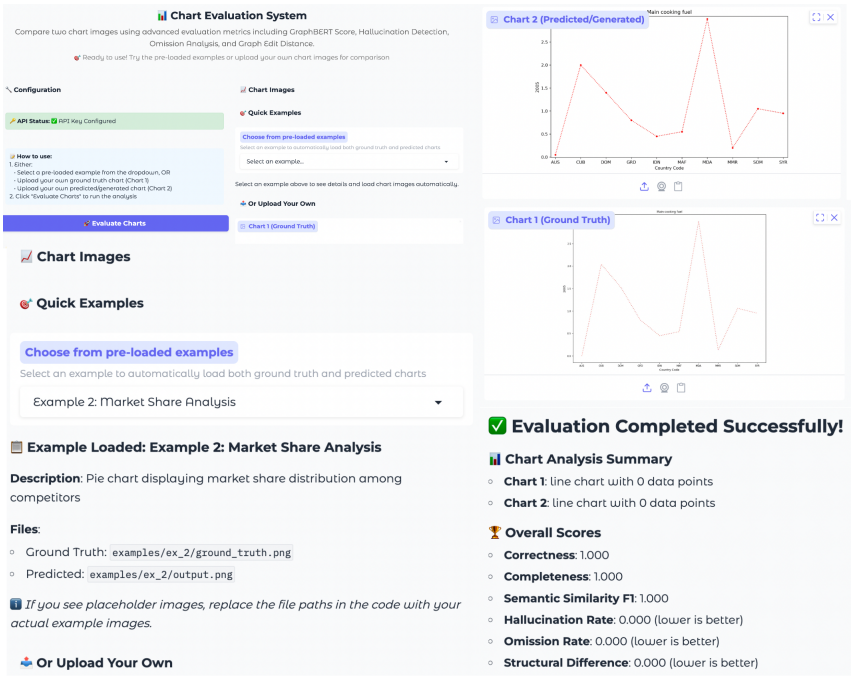

Figure 2: Demo App UI for ChartEval(chartEval.ai)

Metric	Score		
Model Used	Claude	Hallucination Rate	0.333
Correctness (Precision)	1.000	Hallucination Count	4
Completeness (Recall)	1.000	Omission Rate	0.000
F1 Score	1.000	Omission Count	0

Figure 3: ChartEval Example

the structural differences between compared charts.

System License: ChartEval is a proprietary system developed for research experimentation, and is not intended for any commercial purposes.

Usage Scenario Example: Figure 3 shows an example of our tool usage where a user can upload a chart image edited based on user request - "Add a data point (30,25,90) on the line chart" and evaluates it against the reference desired chart. user generated chart erroneously has added more than one data point which is accurately captured by ChartEval as part of H-rate. Further, axis rotation and deviation in rendering quality due to different software does not affect our evaluation system.

5 Experiments - User Study

Datasets: Table 1 summarizes our proposed ChartGen benchmark that comprises of four chart generation datasets spanning three real-world tasks: instruction-based chart image editing, chart re-drawing, and text-to-chart generation.

(1) ChartCraft (Yan et al., 2024) is a dataset of synthetically generated line and bar charts covering style, layout, format, and data-centric edits. We evaluate the language-based chart editing task to modify plot attributes based on user's intent while preserving the integrity of the original plot.

(2) ChartMimic (Shi et al., 2025) contains human-curated visualization from academic documents and scientific papers, covering 22 common chart types. ChartMimic evaluates two tasks: (a) Direct Mimic, where models generate code to reproduce a given chart, and (b) Customized Mimic, where models generate code incorporating new data while preserving the original chart's design.

(3) ChartX (Xia et al., 2024) contains synthetic chart images for re-drawing tasks, where models generate Python code and compare rendered outputs against ground-truth charts.

(4) Text2Chart31 (Pesaran Zadeh et al., 2024) provides chart data across 31 unique plot types, including 3D, volumetric, and gridded charts. We evaluate the description-to-chart task, where each input sample consists of an input textual description and corresponding reference chart.

Settings: We use GPT-4V and Claude-3.7 for ChartSceneParse prompting; GPT-4o, Claude Sonnet-3.5, and Qwen2.5-VL:32b for chart gen-

Metric	Model	ChartCraft						ChartMimic						ChartX						Text2Chart31					
		SSIM	SCRM	GPT-Sc	CB	O-G	O-C	SSIM	SCRM	GPT-Sc	CB	O-G	O-C	SSIM	SCRM	GPT-Sc	CB	O-G	O-C	SSIM	SCRM	GPT-Sc	CB	O-G	O-C
Correct.	GPT-4o	0.09	0.13	0.25	0.34	**0.76***	0.69	0.11	0.15	0.27	0.29	**0.79***	0.72	0.24	0.29	0.33	0.38	0.84	**0.85***	0.18	0.19	0.25	0.25	0.76	**0.78***
	Sonnet-3.5	0.10	0.15	0.23	0.33	**0.75***	0.70	0.13	0.16	0.24	0.31	**0.77***	0.73	0.25	0.27	0.31	0.37	**0.86***	0.86	0.19	0.19	0.26	0.27	0.75	**0.79***
	Qwen2.5-VL	0.15	0.18	0.28	0.36	**0.79***	0.79	0.17	0.19	0.26	0.37	0.79	**0.80***	0.28	0.29	0.33	0.39	0.88	**0.89***	0.22	0.23	0.28	0.29	0.78	**0.79***
Compl.	GPT-4o	0.10	0.14	0.23	0.31	**0.74***	0.70	0.12	0.12	0.28	0.30	**0.76***	0.71	0.22	0.26	0.31	0.35	0.81	**0.82***	0.20	0.19	0.22	0.24	0.75	**0.76***
	Sonnet-3.5	0.08	0.12	0.21	0.28	0.70	**0.72***	0.10	0.15	0.21	0.29	0.75	**0.77***	0.23	0.24	0.28	0.33	0.82	**0.84***	0.21	0.22	0.23	0.35	0.67	**0.72***
	Qwen2.5-VL	0.19	0.19	0.24	0.35	0.76	**0.77***	0.18	0.20	0.23	0.34	0.73	**0.75***	0.27	0.28	0.34	0.36	0.81	**0.82***	0.24	0.25	0.28	0.28	0.76	**0.78***
H-Rate	GPT-4o	0.07	0.11	0.15	0.18	0.45	**0.48***	0.06	0.13	0.15	0.24	0.42	**0.45***	0.05	0.13	0.20	0.24	0.50	**0.52***	0.09	0.12	0.18	0.21	0.55	**0.56***
	Sonnet-3.5	0.08	0.12	0.15	0.19	0.49	**0.51***	0.08	0.17	0.19	0.28	0.48	**0.52***	0.10	0.15	0.19	0.35	0.54	**0.56***	0.11	0.14	0.20	0.24	0.59	**0.60***
	Qwen2.5-VL	0.10	0.12	0.14	0.19	0.40	**0.45***	0.09	0.14	0.19	0.27	**0.45***	0.45	0.08	0.15	0.22	0.27	0.52	**0.54***	0.08	0.14	0.13	0.23	0.53	**0.57***
O-Rate	GPT-4o	0.13	0.15	0.19	0.22	0.51	**0.54***	0.08	0.11	0.18	0.21	0.54	**0.58***	0.12	0.14	0.23	0.28	0.48	**0.51***	0.08	0.11	0.19	0.20	0.51	**0.55***
	Sonnet-3.5	0.14	0.17	0.20	0.24	0.54	**0.56***	0.10	0.13	0.21	0.23	0.55	**0.59***	0.14	0.15	0.22	0.27	0.49	**0.53***	0.12	0.14	0.18	0.22	0.54	**0.56***
	Qwen2.5-VL	0.18	0.19	0.24	0.26	0.54	**0.57***	0.16	0.18	0.23	0.25	0.57	**0.61***	0.19	0.19	0.24	0.29	0.51	**0.54***	0.17	0.18	0.19	0.23	0.55	**0.58***

Table 2: Correlations of ChartEval and existing metrics with human ratings. Correct: Correctness, Compl: Completeness, H: Hallucination, O: Omission, CB: CodeBLEU, GPT-Sc: GPT-Score, O-C(G): Our proposed ChartEval with Claude Sonnet-3.5 (GPT-4) for ChartSceneParse prompting. * indicates statistical significance over GPT-Score ($p \leq 0.005$) under Wilcoxon's Signed Rank test.

eration tasks. More experiment settings in Sec. A.

Baselines Metrics: We compare ChartEval with

(1) GPT-Score (Shi et al., 2025; Xia et al., 2024) uses GPT-4o to compare the candidate and ground truth chart images on a 0-100 scale (normalized to 0-1) based on prompt-based scoring criteria.

(2) SSIM (Wang et al., 2004; Yan et al., 2024) assesses the degree to which the candidate chart visually mirrors the expected outcome, capturing subtle and nuances in the pixel space.

(3) SCRM (Xia et al., 2023) evaluates extracted chart information by converting model-predicted linearized CSV tokens into triplet format, enabling transpose-invariant evaluation of chart data.

(4) CodeBlue (Ren et al., 2020) evaluates the similarity between the predicted and ground truth code for respective charts as in (Pesaran Zadeh et al., 2024). Note that code execution success rate is a standard metric for code generation tasks where unsuccessful executions are assigned a score of 0.

6 Results - User Evaluation

We collected human quality ratings for 4K test charts (1K per dataset) from three annotators, achieving high inter-annotator agreement ($\alpha = \{0.74, 0.82, 0.76, 0.85\}$) across all evaluation metrics. Table 2 presents Pearson correlations between various metrics and these human ratings across different dataset-model combinations. ChartEval consistently demonstrates stronger correlations with human judgments than existing metrics for both proprietary and open-source models across all four datasets. This superior performance indicates that our metric evaluates chart semantics more accurately than baseline approaches. Our analysis reveals several key advantages of ChartEval over existing approaches. Unlike SSIM, our metric avoids over-sensitivity to pixel-level perturbations while successfully capturing meaningful variations in visual attributes such as color schemes, text fonts, and sizing. Code-based metrics like CodeBLEU fail to evaluate spatial misalignment of chart components, which are typically under-specified in generated code. ChartEval offers a decisive advantage over SCRM, which exclusively evaluates underlying data accuracy while ignoring spatial layout and visual design features. Our approach surpasses GPT-Score by mitigating subjective biases inherent in prompt-based evaluation methods. ChartEval is the only metric that provides comprehensive evaluation across semantic, visual, and data dimensions simultaneously, establishing itself as a reliable chart quality assessment tool.

Qualitative Examples: Figures 4-6 illustrate ChartEval's performance across different scenarios. Figure 4 demonstrates accurate evaluation of a 2D area plot where ChartEval correctly identifies near-perfect similarity with zero hallucination rates, avoiding the over-penalization issues of pixel-based metrics. Figure 5 shows ChartEval successfully detects a hallucinated data point in the 3D surface plot, with the Correctness score (0.87) appropriately capturing both spatial inaccuracy and color scheme deviation. **Limitation**: Figure 6 reveals a key limitation that ChartEval struggles with low-resolution input images where the underlying LLM (GPT-4V) hallucinates during scene graph parsing, leading to inaccurate evaluation results. This limitation suggests that ChartEval performs optimally with high-quality images and may require preprocessing steps for low-resolution scenarios. Overall, these examples confirm ChartEval provides more nuanced assessment than existing metrics.

7 Conclusion

We introduced ChartEval, an evaluation system that converts chart images into visual scene graphs and compares their graph-based similarity with ground truth. Extensive experiments across chart reconstruction, text-to-chart synthesis, and editing tasks demonstrate the effectiveness of ChartEval as a reliable chart assessment tool. Future work will explore finetuning VLMs on low resolution chart images for better data extraction.

References

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. Technical Report.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Hussam Ghanem and Christophe Cruz. 2024. Enhancing knowledge graph construction: Evaluating with emphasis on hallucination, omission, and graph similarity metrics. In *International Knowledge Graph and Semantic Web Conference*, pages 32–46. Springer.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Fatemeh Pesaran Zadeh, Juyeon Kim, Jin-Hwa Kim, and Gunhee Kim. 2024. Text2Chart31: Instruction tuning for chart generation with automatic feedback. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11459–11480, Miami, Florida, USA. Association for Computational Linguistics.

Shuo Ren, Daya Guo, Shuai Lu, Long Zhou, Shujie Liu, Duyu Tang, M. Zhou, Ambrosio Blanco, and Shuai Ma. 2020. Codebleu: a method for automatic evaluation of code synthesis. *ArXiv*, abs/2009.10297.

Swarnadeep Saha, Prateek Yadav, Lisa Bauer, and Mohit Bansal. 2021. Explagraphs: An explanation graph generation task for structured commonsense reasoning. *arXiv preprint arXiv:2104.07644*.

Arvind Satyanarayan, Ryan Russell, Jane Hoffswell, and Jeffrey Heer. 2016. Reactive vega: A streaming dataflow architecture for declarative interactive visualization. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*.

Chufan Shi, Cheng Yang, 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. 2025. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation. *International Conference on Learning Representations*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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

Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Peng Ye, Min Dou, Botian Shi, and 1 others. 2024. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *arXiv preprint arXiv:2402.12185*.

Pengyu Yan, Mahesh Bhosale, Jay Lal, Bikhyat Adhikari, and David Doermann. 2024. Chartreformer: Natural language-driven chart image editing. In *International Conference on Document Analysis and Recognition*, pages 453–469. Springer.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A Settings

We conduct experiments using GPT-4o (Hurst et al., 2024), Claude Sonnet-3.5 (Anthropic, 2024), and Qwen2.5-VL:32b (Bai et al., 2025) for chart generation/editing. We render the chart code generated by these models and compare the resulting visualizations against ground truth charts. We use NVIDIA A100 GPUs for Qwen2.5, and APIs for rest. We experiment with GPT-4V and Claude-3.7 for `ChartSceneParse` prompting.

B Qualitative Examples

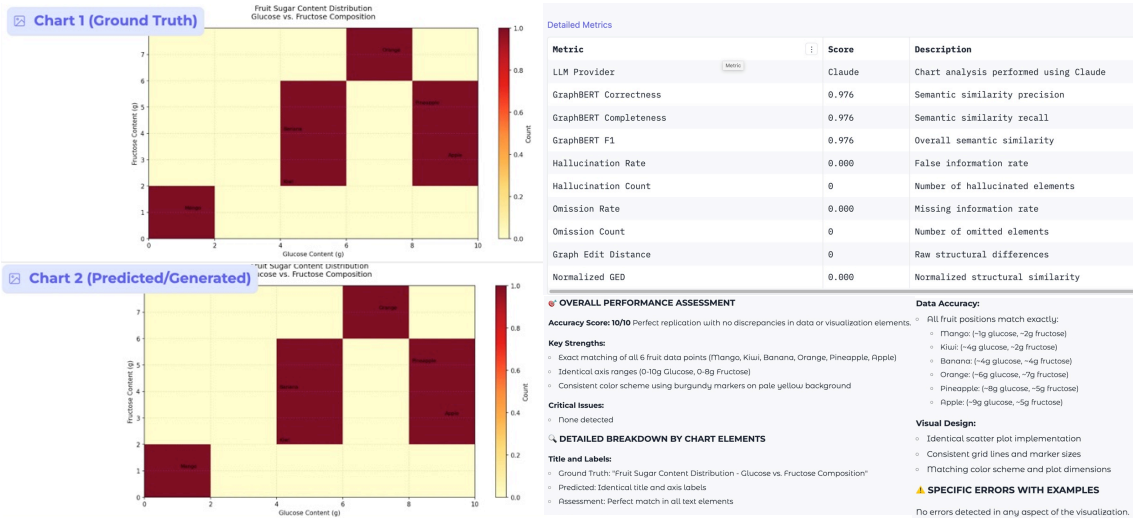

Figure 4: Example of 2D area plot generated by GPT-4o. ChartEval correctly identifies near-perfect chart similarity with zero hallucination or omission rates, demonstrating accurate evaluation of high-quality generated charts.

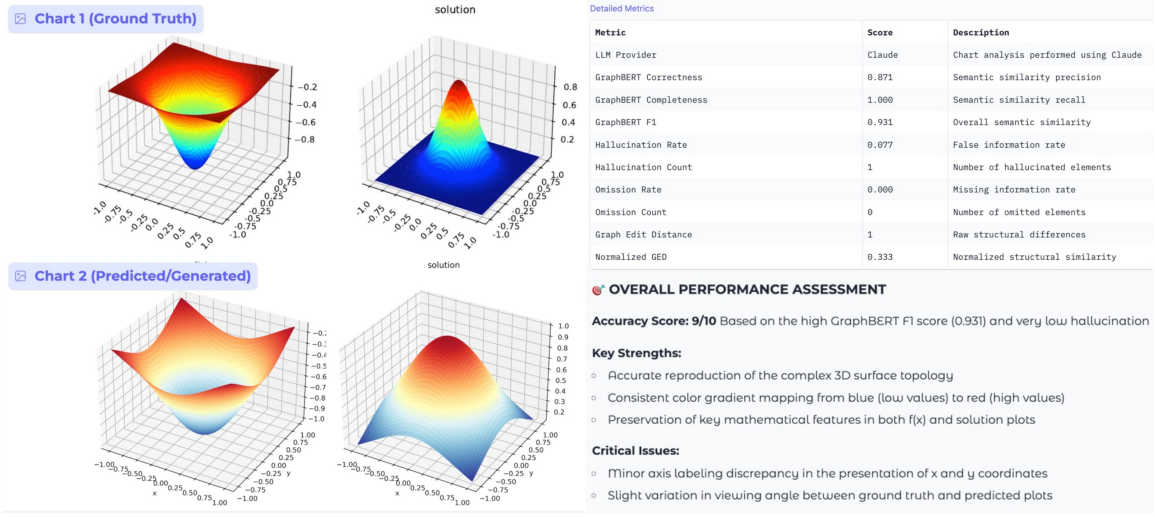

Figure 5: Example of 3D surface plot where ChartEval successfully identifies a hallucinated data point causing curvature distortion. Graph-BERTScore Correctness also accurately detects deviation in the color scheme (Correctness score = 0.87).

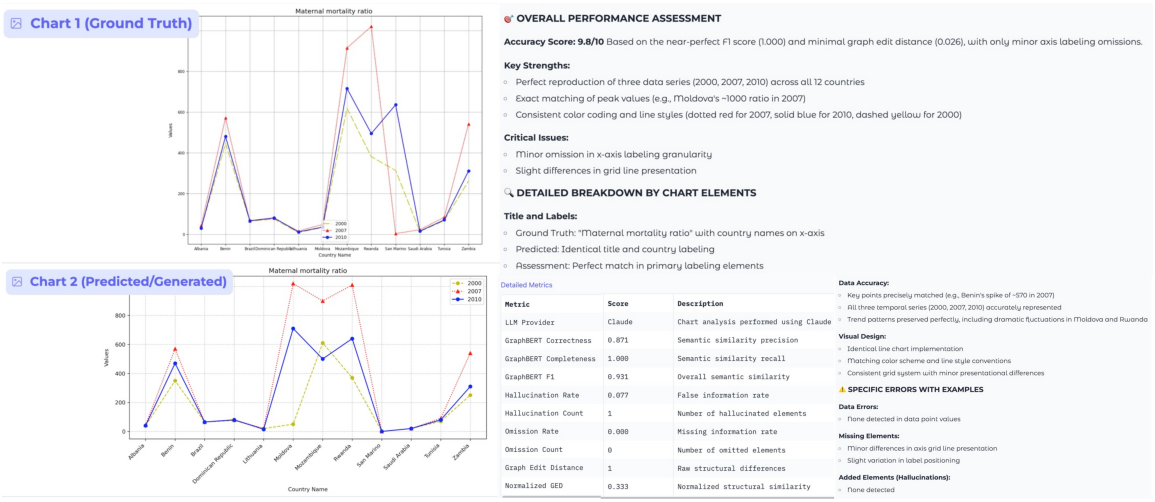

Figure 6: Example of ChartEval limitation on low-resolution images. The predicted chart contains significant data hallucinations and omissions that ChartEval fails to detect due to image quality constraints. Low-resolution inputs cause the underlying LLM (GPT-4V) to hallucinate during scene graph parsing, leading to inaccurate evaluation results.