ChartGalaxy: A Dataset for Infographic Chart Understanding and Generation

Zhen Li[1*] **Yukai Guo**[1*] **Duan Li**[1*] **Xinyuan Guo**[1*] **Bowen Li**[1*] **Lanxi Xiao**[1] **Shenyu Qiao**[1]
Jiashu Chen[1] **Zijian Wu**[1] **Hui Zhang**[1] **Xinhuan Shu**[2] **Shixia Liu**[1†]

[1]Tsinghua University [2]Newcastle University

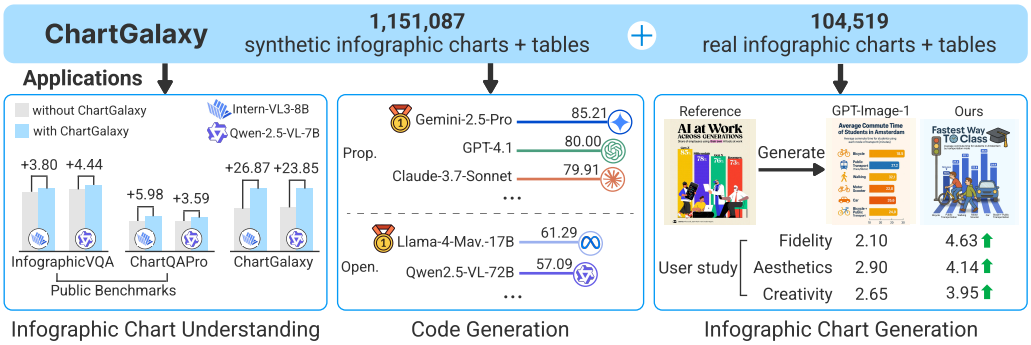


Figure 1: ChartGalaxy, a million-scale dataset of synthetic and real infographic charts with data tables, supporting applications in infographic chart understanding, code generation, and chart generation.

Abstract

Infographic charts are a powerful medium for communicating abstract data by combining visual elements (*e.g.*, charts, images) with textual information. However, their visual and structural richness poses challenges for large vision-language models (LVLMs), which are typically trained on plain charts. To bridge this gap, we introduce ChartGalaxy, a million-scale dataset designed to advance the understanding and generation of infographic charts. The dataset is constructed through an inductive process that identifies 75 chart types, 330 chart variations, and 68 layout templates from real infographic charts and uses them to create synthetic ones programmatically. We showcase the utility of this dataset through: 1) improving infographic chart understanding via fine-tuning, 2) benchmarking code generation for infographic charts, and 3) enabling example-based infographic chart generation. By capturing the visual and structural complexity of real design, ChartGalaxy provides a useful resource for enhancing multimodal reasoning and generation in LVLMs.

Code: https://github.com/ChartGalaxy/ChartGalaxy
Data & Dataset Card: https://huggingface.co/datasets/ChartGalaxy/ChartGalaxy

1 Introduction

Infographic charts are widely recognized as an effective form for communicating data and are commonly used in news media, business, and education [1, 2]. By integrating visual elements—charts,

*Equal contribution.
†Corresponding author.

Preprint. Under review.

imagery such as pictograms and metaphorical graphics—with textual information, they present abstract data in a manner that is both engaging and easy to understand, making data more accessible to a broad audience. Despite their effectiveness for human audiences, foundation models, such GPT-4 [3], Gemini [4], and LLaVA [5], face considerable challenges in automatically understanding infographic charts. The intricate interplay between visual and textual elements, diverse layout styles, and the need for cross-modal semantic reasoning pose significant difficulties. Moreover, automatically generating high-quality infographic charts also remains an open challenge. While human designers can create visually diverse and semantically rich infographic charts, this process is time-consuming and requires expertise. Meanwhile, AI-generated charts often suffer from issues such as low data fidelity, modest visual quality, limited diversity, and a lack of coherence across modalities. This highlights the critical need for a comprehensive dataset of infographic charts that can support the development of deep learning models capable of both automatic understanding and generation. However, existing efforts focus on constructing datasets that are mostly limited to plain charts, failing to capture the diverse range of design styles and layouts that are key characteristics of infographic charts. This limits the ability of the trained models to generalize across different real-world applications where infographic charts are commonly used.

To address this limitation, we build ChartGalaxy, a million-scale dataset of high-quality real and synthetic infographic charts to facilitate automated understanding and generation. As shown in Fig. 2, we build ChartGalaxy in two steps: 1) collecting real infographic charts; 2) programmatically creating synthetic infographic charts. The real infographic charts are collected from 19 reputable chart-rich websites, such as *Pinterest* [6], *Visual Capitalist* [7], *Statista* [8], and *Information is Beautiful* [9]. The synthetic infographic charts are created following an inductive structuring process [10]. Specifically, we identify design patterns grounded in real infographic charts, including **75 chart types** (*e.g.*, bar charts), **330 chart variations** that reflect different visual element styles, and **68 layout templates** that define spatial relationships among elements. Based on these patterns, we then programmatically generate synthetic ones. The core of the generation is a human-in-the-loop pipeline that iteratively extracts and expands layout templates from real infographic charts using a detection model trained on synthetic infographic charts.

The final ChartGalaxy dataset includes 1,151,087 programmatically created infographic charts and 104,519 real infographic charts. It is characterized by two key features. First, the high-quality infographic designs and associated templates from these reputable websites ensure a rich diversity in design styles and structural complexity. Second, each infographic chart, whether real or synthetic, is paired with the tabular data used to create it, enabling a clear mapping between the data and its visual representation. Together, these features make ChartGalaxy a useful dataset for training and evaluating LVLMs designed for the automatic understanding and generation of infographic charts. We demonstrate the utility of ChartGalaxy through three representative applications, each highlighting a distinct aspect of its value (Fig. 1). First, to evaluate and improve the ability of foundation models to understand infographic charts, we introduce a dataset for infographic chart understanding through the task of visual question answering (VQA). Second, to assess the capacity of models to generate executable representations of complex visual layouts, we present a benchmark for infographic chart code generation. Third, to explore the use of ChartGalaxy in creative content generation, we develop an example-based infographic chart generation method.

The main contributions of our work include:

- A pipeline for programmatically creating high-quality synthetic infographic charts based on the extracted layout templates from real designs.
- A comprehensive dataset comprising a large collection of representative and diverse real and synthetic infographic charts paired with tabular data.
- Three applications for showcasing the utility of our dataset in infographic chart understanding, code generation, and example-based infographic chart generation.

2 Related Work

Early efforts in chart dataset construction primarily focus on building collections of **plain charts** to support chart understanding and generation [11]. These datasets can be further categorized into three types based on their sources: synthetic datasets, web-crawled datasets, and mixed datasets. **Synthetic datasets** are programmatically generated, using tabular data drawn from probability distributions [12,

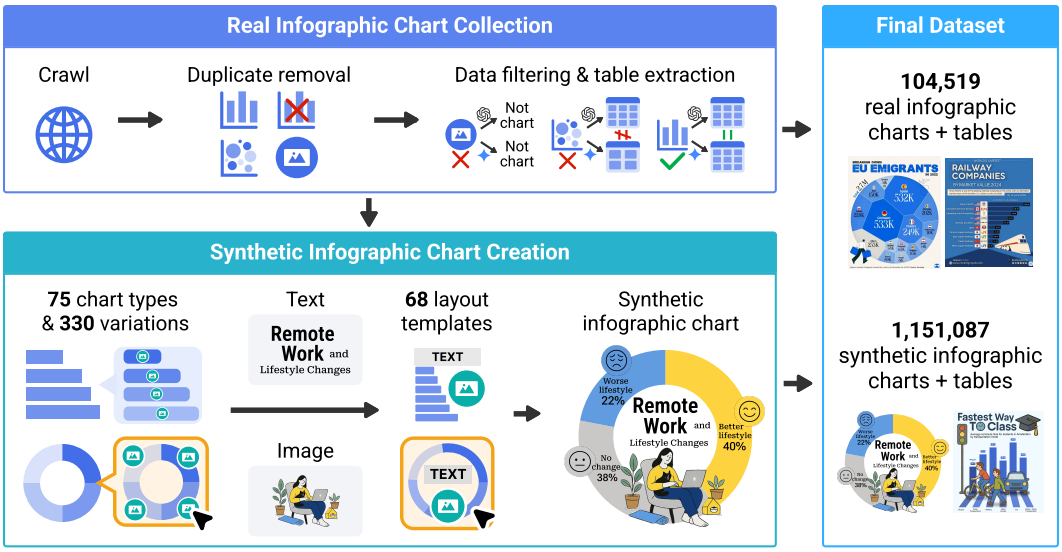


Figure 2: Overview of our dataset construction method.

13], crawled from online data sources [11, 14, 15, 16, 17, 18], or simulated from large language models [16, 19, 20, 21]. While this method enables large-scale dataset construction, the controlled generation process often results in a limited diversity of chart types and visual styles, which reduces generalizability to more varied real-world scenarios. To improve diversity, **web-crawled datasets** have been introduced, which collect charts from chart-sharing websites [22, 23, 24, 25, 26, 27, 28, 29], charting library galleries such as Matplotlib [30, 31], and publications in academic repositories such as ArXiv [32, 33, 34]. These datasets capture a broader variety of human-designed chart styles, but their overall size is often limited due to the time-consuming nature of manual verification and annotation. To overcome the scale limitations while preserving diversity, **mixed datasets** are then introduced. These datasets merge web-crawled charts with synthetically generated ones in a two-step workflow: 1) collect real charts from the internet and 2) manually synthesize additional charts that follow the same real-world design patterns [35, 36, 37, 38, 39]. This hybrid strategy ensures real-world coverage while increasing dataset size.

In contrast, **infographic charts** remain underrepresented in the aforementioned datasets. This creates challenges in evaluating and developing LVLMs' capabilities on infographic charts [40]. To bridge this gap, recent efforts have focused on building specific datasets for infographic charts. InfographicVQA makes an initial effort by searching "infographics" on the internet and scraping 5,485 infographics [41]. More recently, ChartQAPro provides a more challenging benchmark comprising 190 infographic charts, 258 dashboards, and 893 plain charts [40]. However, these datasets are limited in scale. Moreover, synthesizing infographic charts is challenging, given the intricate interplay between visual and textual elements. To overcome these limitations, we develop an automatic infographic chart generation method that synthesizes infographic charts by leveraging the layout templates and chart variations extracted from real designs.

3 Dataset Construction Method

3.1 Method Overview

Fig. 2 presents an overview of our method, which includes two stages: **real infographic chart collection** and **synthetic infographic chart creation**.

The chart collection stage aims to collect real infographic charts by crawling 19 chart-rich websites, such as *Pinterest* [6], *Visual Capitalist* [7], *Statista* [8]. The full list of these websites is provided in Supp. A. To ensure data quality, we remove duplicate images using perceptual hashing [42] and CLIP similarity [43]. We then extract the data table for each chart using GPT-4o-mini and Gemini-

2.0-Flash independently, retaining only those with consistent tabular outputs. This results in 104,519 real infographic charts with corresponding tables. The chart creation stage follows an inductive structuring process that extracts design patterns, such as layout templates and chart variations, from real infographic charts and then uses these patterns to programmatically create high-quality synthetic charts. It includes three steps: 1) identifying chart types and their variations, 2) extracting layout templates, and 3) creating synthetic infographic charts with the identified design patterns.

3.2 Chart Type and Variation Identification

We first summarize 75 chart types observed in the collected real infographic charts, drawing on two existing taxonomies: Data Viz Project [44] and Datylon [45]. For each chart type, we extract chart variations featuring diverse visual styles, such as element shapes and icon placement. This results in 330 chart variations in total. The full lists of chart types and variations are provided in Supp. B.1 and B.2. We implement these chart types and variations using the expressive D3.js [46], which supports visual features unavailable in libraries like Matplotlib or Seaborn.

3.3 Layout Template Extraction

A layout template defines the spatial relationships among the text and visual elements in an infographic chart. Example templates are shown on the bottom-left corner of each chart in Fig. 3. We adopt a human-in-the-loop pipeline to initialize and expand these templates from real infographic charts.

Initialization Three co-authors manually annotate the bounding boxes of the text, image, and chart in 1,500 real infographic charts sampled from two high-quality sources: *Statista* [8] (clean, minimalist designs) and *Visual Capitalist* [7] (denser and more pictorial designs). From these annotations, we summarize an initial set of 55 layout templates that capture elements' relative positions (*e.g.*, title on the top-left, chart on the bottom-right) and pairwise overlaps (overlapping or not).

Expansion To ensure the coverage and diversity of templates, we build a detection model to analyze the unlabeled real infographic charts and systematically expand the template set. Using the initial templates, we programmatically create 120,000 synthetic infographic charts (Sec. 3.4), each with annotated bounding boxes. We then develop a detection model by fine-tuning InternImage [47] along with the DINO [48] detector on these synthetic charts. Applied to real unlabeled charts, this model detects chart and image regions, while text is extracted using PP-OCRv4 [49]. We then compare the detected layouts with the existing templates using LTSim [50], a state-of-the-art method for measuring layout similarity. Layouts with low similarity scores are flagged as potential new templates. Next, we cluster these layouts using k-means ($k = 50$) and manually examine the cluster centroids to identify distinct layouts. This process yields 13 additional layout templates, expanding the set to 68 templates in total. The full list is provided in Supp. C.

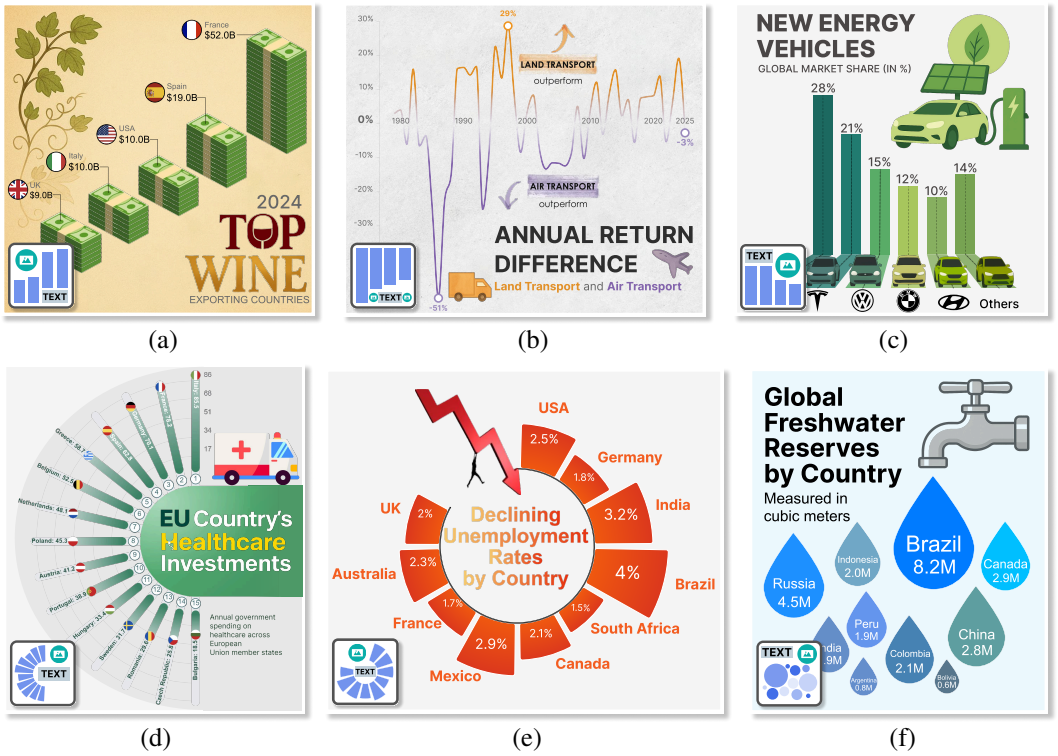
3.4 Template-based Infographic Chart Creation

The creation process involves three steps: 1) curating data tables; 2) generating/recommending elements based on the data table; and 3) optimizing the layout based on the selected template.

Tabular data curation To enhance data diversity for chart generation, we build a rich repository of real and synthetic tabular data. For real data, we collect 200,085 tables from well-established sources, including VizNet [51], UN Data [52], Our World in Data [53], and Papers with Code [54]. For synthetic data, we generate 98,483 tables with Gemini-2.0-Flash following Han *et al.*'s method [20]. To facilitate downstream processing, we also complement each table with a topic (*e.g.*, "US election," "NBA play-offs") extracted by Gemini-2.0-Flash and several data facts (*e.g.*, trends, comparisons) [55].

Element generation/recommendation For each data table, we generate/recommend key elements of the infographic chart, including 1) text, 2) image, and 3) chart.

Text. The titles and subtitles are generated using a retrieval-augmented prompting strategy to reflect real-world usage. Specifically, we use Sentence-BERT [56] to retrieve the three most relevant real infographic charts according to the data topics and data facts. Using these retrieved examples as in-context references, we prompt Gemini-2.0-Flash to generate titles and subtitles aligned with the tabular data.



(a) (b) (c)

(d) (e) (f)

Figure 3: Examples of synthetic infographic charts in ChartGalaxy. The bottom-left illustration on each infographic chart shows the corresponding layout template.

Image. To recommend images for infographic charts, we construct an image repository and retrieve semantically relevant images based on the associated tabular data. Images are selectively collected from publicly available resources such as Icon645 [57], Flaticon [58], and Noun Project [59]. We apply heuristic filters to remove low-quality images, excluding those with low ink ratios, poor resolution, or extreme aspect ratios. Using Gemini-2.0-Flash, we then generate descriptive keywords and captions for each image that depict its content and visual style. Images that are overly literal, visually cluttered, or lack symbolic clarity are discarded. This process results in a curated repository of 681,459 high-quality images. To retrieve relevant images for each infographic chart, we compute the similarity between the image keywords and the generated chart title using Sentence-BERT [56] embeddings.

Chart. Chart generation proceeds in two steps: finding suitable chart types and rendering with a specific chart variation. Chart types are determined by analyzing the data attributes (*e.g.*, categorical, numerical) and characteristics (*e.g.*, scales). For example, a data table with one categorical column and two numerical columns may be mapped to a scatter plot. When multiple chart types are suitable, we prompt Gemini-2.0-Flash to select the optimal one based on data-chart compatibility [60]. A full list of mapping rules and prompts used for chart selection is provided in Supp. B.3. Once the chart type is selected, we choose a specific variation using an adaptive sampling strategy that favors underrepresented variations to ensure distributional balance. For variations requiring additional images (*e.g.*, Figs. 3(a), 3(c) and 3(d)), we retrieve relevant ones from our curated image repository using the above image recommendation method. Finally, we apply semantically resonant color palettes for chart rendering based on the generated chart titles and subtitles. These palettes are extracted from the collected real infographic charts following Liu *et al.*'s method [61]. If a selected palette contains fewer colors than required, Chen *et al.*'s method is used to supplement additional harmonic and discriminable colors [62].

Layout optimization Previous research has shown that a compact layout with appropriate white space enhances both visual appeal and data clarity [63]. Consequently, we aim to select the template with the highest ink ratios while preserving readability. To this end, we first filter out templates that are incompatible with the generated elements (*e.g.*, unintended overlaps between elements). For each

remaining template, we initialize the positions for the set of elements, optimize the layout to reduce unnecessary white space, and compute the ink ratio. The template with the highest ink ratio is then selected for chart creation.

$$\max_{\mathcal{E}} \; |\cup_i e_i| \; / \; |f(\cup_i e_i)|, \qquad \text{s.t.} \quad d(\partial e_i, \partial e_j) \geq p, \; \forall \, i \neq j. \tag{1}$$

Here, e_i is the pixel set of an element, $f(e)$ denotes the pixel set within the tight-fitting bounding box of e, and $\mathcal{E}$ is the set of elements. We enforce that a minimum pairwise distance d between element contours ∂e_i and ∂e_j is larger than a given threshold p. This layout optimization is formulated as a constrained packing problem and solved by grid search [64]. Fig. 3 shows representative examples of the synthetic charts. For more examples, please refer to Supp. C.

3.5 Data Statistics

ChartGalaxy contains 1,151,087 programmatically generated infographic charts and 104,519 real ones, covering 75 chart types, 330 chart variations, and 68 layout templates. Among the 75 chart types, the most frequently occurring ones are horizontal bar charts (11.7%), vertical bar charts (4.9%), and scatterplots (3.5%), reflecting their common usage in infographic-style presentations. Each chart type has up to 26 variations, capturing a wide range of styles in icon placement, color encoding, and data-label associations. We also observe variation in the frequency of layout template usage. For example, the template in Fig. 3(f) is the most frequently used, accounting for 6.9% of the synthetic charts. These differences are influenced by the structural flexibility of each template, their compatibility with various chart types, and their prevalence in real-world use cases. For more information on chart types, chart variations, and templates, please refer to Supp. B and Supp. C. Each infographic chart in the dataset is associated with tabular data, providing rich supervision for training and evaluation tasks.

4 Experiments

4.1 Instruction Dataset for Infographic Chart Understanding

In this experiment, we construct an instruction dataset with ChartGalaxy to enhance model capabilities on infographic chart understanding. We validate its usefulness by fine-tuning two open-source LVLMs, demonstrating clear improvements on both public benchmarks and our independent evaluation set.

Dataset construction To improve LVLMs' data comprehension and visual understanding of infographic charts, we construct an instruction dataset comprising 443,455 question-answer pairs based on 70,248 charts randomly sampled from ChartGalaxy. The questions are classified into three types: 1) **Text-based reasoning**. We incorporate well-established question types from prior work, including open-form questions [40] and template-based questions [38]. These questions cover data identification (DI), data comparison (DC), data extraction with condition (DEC), and fact checking (FC). 2) **Visual-element-based reasoning**. We extend beyond purely text-based reasoning questions by incorporating visual elements from charts, such as icons (*e.g.*, "What was the wine export value of 🇫🇷 in 2024?"). These questions require models to associate visual elements with their corresponding data values, thus testing their ability to conduct more complex cross-modal reasoning. 3) **Visual understanding**. This type includes style detection (SD), visual encoding analysis (VEA, *e.g.*, "What data dimension is encoded using different colors in this infographic chart?"), and chart classification (CC). The three types of questions evaluate a model's ability to interpret visual design elements and underlying structural representations of the data. Detailed prompts and methods for generating question-answer pairs are provided in Supp. E.1. We also construct an independent, human-verified evaluation set containing 2,176 charts with 4,975 question-answer pairs for systematic evaluation.

Experimental setup We fine-tune two representative open-source LVLMs, InternVL3-8B [65] and Qwen2.5-VL-7B [66]. Training details are reported in Supp. D.1. Our evaluation benchmark consists of two parts: 1) public benchmarks including InfographicVQA [41], which focuses on general infographics with only a subset being infographic charts, and ChartQAPro [40], which covers various chart types; and 2) the aforementioned independent evaluation set of 2,176 charts with 4,975

Table 1: Performance on public benchmarks w/ and w/o ChartGalaxy.

Model	InfographicVQA	ChartQAPro
InternVL3-8B	76.19	38.15
+ ChartGalaxy	79.99	**44.13**
(+)	(+3.80)	(+5.98)
Qwen2.5-VL-7B	78.59	37.97
+ ChartGalaxy	**83.03**	41.56
(+)	(+4.44)	(+3.59)

Table 2: Performance on our independent evaluation set w/ and w/o ChartGalaxy.

Model	Text-Based Reasoning				Visual-Element-Based Reasoning				Visual Understanding			Overall
	DI	DC	DEC	FC	DI	DC	DEC	FC	SD	VEA	CC	
InternVL3-8B	85.36	55.24	51.66	75.80	33.32	18.91	37.62	61.58	30.56	50.57	73.03	53.20
+ ChartGalaxy	91.67	74.39	75.14	**89.26**	**69.12**	**42.79**	58.57	**80.23**	**91.05**	**91.35**	**99.39**	80.07
(+)	(+6.31)	(+19.15)	(+23.48)	(+13.46)	(+35.80)	(+23.88)	(+20.95)	(+18.65)	(+60.49)	(+40.78)	(+26.36)	(+26.87)
Qwen2.5-VL-7B	87.45	66.32	64.44	78.53	40.76	30.65	46.00	53.95	28.70	50.08	70.91	56.50
+ ChartGalaxy	**93.28**	**80.98**	**86.31**	87.34	66.15	39.80	**72.38**	79.38	87.65	90.86	98.18	**80.35**
(+)	(+5.83)	(+14.66)	(+21.87)	(+8.81)	(+25.39)	(+9.15)	(+26.38)	(+25.43)	(+58.95)	(+40.78)	(+27.27)	(+23.85)

question-answer pairs specifically targeting infographic
charts. For the evaluation metrics, we follow previous work on chart question answering [40], using relaxed accuracy with a 5% margin for numerical answers, ANLS for textual answers, and exact matching for multiple-choice questions.

Results and analysis Tables 1 and 2 show the evaluation results on the public benchmarks and our evaluation set. After fine-tuning with ChartGalaxy, both models demonstrate improved performance gains across all question types. On the public benchmarks (Table 1), InternVL3 improves performance by 3.80% on InfographicVQA and 5.98% on ChartQAPro, while Qwen2.5-VL shows a 4.44% gain on InfographicVQA and a 3.59% improvement on ChartQAPro. On our evaluation set (Table 2), both models show consistent improvements across all question types, with overall gains of +26.87% for InternVL3 and +23.85% for Qwen2.5-VL. The most notable improvements are observed in the visual understanding questions, with increases of up to +60.49% for style detection and +40.78% for visual encoding analysis. These results indicate that existing pre-training routines may underrepresent questions involving chart visual styles and data encoding, an area our dataset helps to supplement. Performance also improves across the text-based and visual-element-based reasoning questions. Qwen2.5-VL performs well on text-based reasoning, while InternVL3 shows relatively stronger gains on visual-element-based reasoning.

4.2 Benchmarking Infographic Chart Code Generation

This experiment presents a benchmark to assess LVLMs' code generation for infographic charts.

Benchmark construction The benchmark is designed to evaluate the Direct Mimic task [39], where an LVLM is prompted to generate the D3.js code for a given infographic chart image. Due to variation in coding styles and implementation strategies [67], directly comparing code quality is challenging. Therefore, we evaluate the rendered output instead of the code itself. Specifically, we render the output as both an SVG and a PNG: the SVG enables fine-grained analysis, as it contains precise information about visual and textual elements (*e.g.*, positions, colors), while the PNG supports direct visual comparison. To support this task, we sample 500 infographic charts from ChartGalaxy, covering diverse chart types, variations, and layouts. Each chart is paired with a ground-truth triplet: a PNG image, an SVG, and the corresponding tabular data. Benchmark details are provided in Supp. D.2.

Following previous benchmarks [67, 39], we measure the similarity between the ground-truth chart and the one rendered by the generated code at two levels: a **high-level score** (overall visual similarity judged by GPT-4o with the PNG images) and a **low-level score** (average similarity across fine-grained SVG elements). To compute the low-level score, we parse the SVG elements from the rendered chart and the ground-truth one and match them based on attributes such as tag types and positions. This matching is formulated as a linear assignment problem and solved using the Jonker-Volgenant algorithm [68, 67, 69]. Based on the matching results, we compute a low-level score by averaging six metrics: area, text, image, color, position, and size. The area metric captures the ratio of matched element area to the total element area. The text and image metrics assess the similarity of generated text and image elements, respectively. The color, position, and size metrics evaluate visual consistency in these attributes among matched elements. Details of the evaluation metrics are provided in Supp. D.2. As in [39], we also calculate the **overall** score as the average of the high-level and low-level scores, ranging from 0 to 100. Notably, if the code fails to render the chart, both scores are set to 0.

Experimental setup We benchmark 17 widely used LVLMs, including 12 proprietary ones and 5 open-source ones, as shown in Table 3. Model configurations and detailed prompts are provided in Supp. D.2 and Supp. E.2, respectively.

Table 3: Performance comparison of 17 LVLMs on our proposed code generation benchmark, reporting the code execution success rate (Exec. Rate), low-level, high-level, and overall scores.

Model	Exec. Rate	Low-Level							High-Level	Overall
		Area	Text	Image	Color	Position	Size	Avg.	GPT-4o	
Proprietary										
Gemini-2.5-Pro [70]	**100.00**	**90.72**	**95.69**	86.37	**87.67**	**89.23**	**69.05**	**86.45**	**83.97**	**85.21**
GPT-4.1 [71]	**100.00**	90.58	91.58	**86.53**	87.52	87.13	55.61	83.16	76.84	80.00
Claude-3.7-Sonnet [72]	**100.00**	88.96	92.39	77.90	84.78	87.57	67.29	83.15	76.66	79.91
GPT-4.1-mini [71]	99.60	88.21	88.31	79.32	86.43	85.61	62.85	81.79	77.59	79.69
OpenAI-o4-mini [73]	98.80	83.13	79.26	67.53	83.93	84.95	64.07	77.14	74.79	75.97
Gemini-2.5-Flash [74]	96.40	84.52	87.02	73.21	77.81	83.01	62.29	77.98	73.12	75.55
OpenAI-o1 [75]	97.20	85.01	78.07	80.28	81.70	82.35	60.76	78.03	71.35	74.69
OpenAI-o3 [73]	92.40	83.84	82.43	72.15	78.79	81.63	63.44	77.05	71.38	74.22
GPT-4o [76]	99.00	74.86	63.54	34.12	76.60	80.12	56.27	64.25	67.10	65.67
GPT-4.1-nano [71]	**100.00**	74.97	59.94	36.06	69.26	75.10	47.69	60.50	59.62	60.06
Doubao-1.5-Vision-Pro [77]	97.20	59.05	48.08	36.63	60.09	66.02	40.00	51.65	42.58	47.11
Moonshot-v1-Vision [78]	95.20	60.86	45.54	33.68	60.86	63.33	39.81	50.68	38.11	44.39
Open-Source										
Llama-4-Maverick-17B [79]	**99.60**	**76.59**	56.37	59.24	**69.59**	**75.39**	**49.87**	**64.51**	**58.06**	**61.29**
Qwen2.5-VL-72B [66]	92.60	73.50	**63.22**	49.33	65.06	73.14	47.53	61.96	52.21	57.09
InternVL3-78B [65]	95.60	73.36	47.98	**62.19**	65.76	70.35	43.74	60.56	49.58	55.07
Llama-4-Scout-17B [79]	98.20	71.08	51.81	45.27	63.48	69.95	42.28	57.31	46.50	51.91
Qwen2.5-VL-32B [66]	81.60	60.37	49.37	28.19	54.37	61.37	37.60	48.55	44.42	46.48

Results and analysis Table 3 presents the results of the 17 LVLMs on our benchmark. We highlight key findings below. First, among proprietary models, Gemini-2.5-Pro achieves the highest overall score of 85.21. Among open-source models, Llama-4-Maverick-17B performs the best with a score of 61.29, outperforming the proprietary GPT-4.1-nano (60.06). However, it still lags behind top-performing proprietary models. Within the GPT-4.1 series, GPT-4.1 and GPT-4.1-mini achieve nearly identical overall scores (80.00 vs. 79.69). However, GPT-4.1 consistently outperforms GPT-4.1-mini across all individual low-level metrics, except the size metric, where it scores notably lower (55.61 vs. 62.85). This drop suggests that GPT-4.1 may struggle to preserve element size accurately, highlighting the need for further investigation. Additional results and analysis are provided in Supp. D.2.

4.3 Example-based Infographic Chart Generation

This experiment demonstrates how ChartGalaxy can be used to support the generation of infographic charts through layout and style adaptation.

Method We develop an example-based method that transforms user-provided tabular data into an infographic chart, aligning with the layout and visual style of a given example infographic chart. This example is either provided by the user or automatically retrieved from the ChartGalaxy dataset by selecting the chart most relevant to the user-provided tabular data and its column descriptions. The key feature of this method is its ability to generate visually coherent infographic charts by reusing the layout templates of well-designed examples and leveraging powerful detection and vision-language models. To enable this capability, we first apply the detection model described in Sec. 3.3 to detect key elements (*e.g.*, icons, text blocks) in the example infographic chart, from which the layout template is constructed. Then, we use the template-based infographic chart creation method to populate this template with the provided tabular data, generating a new infographic that preserves the example's aesthetic style while incorporating the input data seamlessly.

User study setup We conducted a user study with 16 experts in design or visualization to evaluate the quality of the generated infographic charts using our method and GPT-Image-1 [80], a state-of-the-art image generation model. The evaluation focused on three key metrics: **fidelity** (how accurately the data is represented), **aesthetics** (how appealing the infographic chart is), and **creativity** (how innovative the design is). Experts were asked to rate 30 pairs of infographic charts generated by our method and GPT-Image-1 based on the same tabular data and reference infographic chart, using a

Figure 4: Three examples of infographic charts used in Sec. 4.3. In each example, A is the reference chart, B and C are generated by GPT-Image-1 and our method, respectively, using the same data.

Table 4: Performance comparison between our method and GPT-Image-1 (Mean, [95% CI]).

Method	Fidelity	Aesthetics	Creativity
Ours	4.63, [4.51, 4.75]	4.14, [3.95, 4.33]	3.95, [3.77, 4.13]
GPT-Image-1	2.10, [1.71, 2.50]	2.90, [2.48, 3.33]	2.65, [2.28, 3.03]

five-point Likert scale (1=poor, 5=excellent) for each metric. Fig. 4 shows examples of the generated infographic charts and the reference. The prompt used for GPT-Image-1 is provided in Supp. E.3.

Results and analysis Table 4 shows that our method significantly outperforms GPT-Image-1 across all three metrics based on the Wilcoxon signed-rank test on the user rating data ($p < 0.01$): **fidelity** (average: 4.63 vs. 2.10), **aesthetics** (4.14 vs. 2.90), and **creativity** (3.95 vs. 2.65). Particularly, our method achieves high fidelity by accurately representing data through carefully implemented chart variations. Some scores are slightly lower due to expert preferences for alternative chart types in certain cases. In contrast, GPT-Image-1 often exhibits serious fidelity-related issues, such as incorrect labels, disproportionate representations, and mismatched data elements. In terms of aesthetics and creativity, our method benefits from accurately extracting high-quality layout templates from reference infographic charts and supporting a wider variety of chart types beyond basic chart types, such as bar charts and line charts. By comparison, GPT-Image-1 tends to use basic chart types with limited variations, leading to outputs that are visually simple and monotonous. The detailed analysis is provided in Supp. D.3.

5 Conclusion

We echo the growing interest in multimodal understanding and generation in LVLMs by introducing ChartGalaxy, a million-scale, high-quality dataset of 104,519 real and 1,151,087 synthetic infographic charts. Grounded in real designs, our structured synthesis pipeline enables the scalable creation of diverse infographic charts. By providing aligned data-chart pairs, extracted layout templates, and three representative applications, we aim to advance the development of foundation models capable of interpreting, reasoning, and generating complex infographic charts.

At the same time, we acknowledge the current ChartGalaxy dataset primarily focuses on single-chart infographics, limiting its ability to capture the complexity of multi-chart narratives. As a result, future work should explore generating and analyzing multi-chart infographics, which emphasize storytelling through coordinated visual elements. Additionally, enriching the interplay between text and visuals could further enhance models' capacity for nuanced multimodal understanding.

References

[1] W. Cui, J. Wang, H. Huang, Y. Wang, C.-Y. Lin, H. Zhang, and D. Zhang, "A mixed-initiative approach to reusing infographic charts," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 173–183, 2022.

[2] S. Elaldi and T. Çifçi, "The effectiveness of using infographics on academic achievement: A meta-analysis and a meta-thematic analysis," *Journal of Pedagogical Research*, vol. 5, no. 4, pp. 92–118, 2021.

[3] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[4] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican *et al.*, "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[5] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 34 892–34 916, 2023.

[6] "Pinterest," 2025, accessed: 2025-02-04. [Online]. Available: https://www.pinterest.com

[7] "Visual Capitalist," 2025, accessed: 2025-02-04. [Online]. Available: https://www.visualcapitalist.com

[8] "Statista," 2025, accessed: 2025-05-07. [Online]. Available: https://www.statista.com/

[9] "Information is Beautiful," 2025, accessed: 2025-02-04. [Online]. Available: https://www.informationisbeautiful.net

[10] N. Schadewitz and T. Jachna, "Comparing inductive and deductive methodologies for design patterns identification and articulation," in *Proceedings of the International Design Research Conference: Emerging Trends in Design Research*, 2007, pp. 1–19.

[11] L. Hu, D. Wang, Y. Pan, J. Yu, Y. Shao, C. Feng, and L. Nie, "NovaChart: A large-scale dataset towards chart understanding and generation of multimodal large language models," in *Proceedings of the ACM International Conference on Multimedia*, 2024, pp. 3917–3925.

[12] K. Kafle, B. Price, S. Cohen, and C. Kanan, "DVQA: Understanding data visualizations via question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5648–5656.

[13] S. E. Kahou, V. Michalski, A. Atkinson, Á. Kádár, A. Trischler, and Y. Bengio, "FigureQA: An annotated figure dataset for visual reasoning," *arXiv preprint arXiv:1710.07300*, 2017.

[14] N. Methani, P. Ganguly, M. M. Khapra, and P. Kumar, "PlotQA: Reasoning over scientific plots," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1527–1536.

[15] R. Chaudhry, S. Shekhar, U. Gupta, P. Maneriker, P. Bansal, and A. Joshi, "LEAF-QA: Locate, encode & attend for figure question answering," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 3512–3521.

[16] Z. Xu, S. Du, Y. Qi, C. Xu, C. Yuan, and J. Guo, "ChartBench: A benchmark for complex visual reasoning in charts," *arXiv preprint arXiv:2312.15915*, 2023.

[17] B. Tang, A. Boggust, and A. Satyanarayan, "VisText: A benchmark for semantically rich chart captioning," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2023, pp. 7268–7298.

[18] M. Akhtar, O. Cocarascu, and E. Simperl, "Reading and reasoning over chart images for evidence-based automated fact-checking," in *Findings of the Association for Computational Linguistics*, 2023, pp. 399–414.

[19] R. Xia, B. Zhang, H. Ye, X. Yan, Q. Liu, H. Zhou, Z. Chen, P. Ye, M. Dou, B. Shi *et al.*, "ChartX & ChartVLM: A versatile benchmark and foundation model for complicated chart reasoning," *arXiv preprint arXiv:2402.12185*, 2024.

[20] Y. Han, C. Zhang, X. Chen, X. Yang, Z. Wang, G. Yu, B. Fu, and H. Zhang, "ChartLlama: A multimodal LLM for chart understanding and generation," *arXiv preprint arXiv:2311.16483*, 2023.

[21] X. Zhao, X. Luo, Q. Shi, C. Chen, S. Wang, W. Che, Z. Liu, and M. Sun, "ChartCoder: Advancing multimodal large language model for chart-to-code generation," *arXiv preprint arXiv:2501.06598*, 2025.

[22] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer, "Revision: Automated classification, analysis and redesign of chart images," in *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, 2011, pp. 393–402.

[23] J. Choi, S. Jung, D. G. Park, J. Choo, and N. Elmqvist, "Visualizing for the non-visual: Enabling the visually impaired to use visualization," in *Computer Graphics Forum*, vol. 38, no. 3, 2019, pp. 249–260.

[24] A. Masry, X. L. Do, J. Q. Tan, S. Joty, and E. Hoque, "ChartQA: A benchmark for question answering about charts with visual and logical reasoning," in *Findings of the Association for Computational Linguistics*, 2022, pp. 2263–2279.

[25] S. Kantharaj, X. L. Do, R. T. Leong, J. Q. Tan, E. Hoque, and S. Joty, "OpenCQA: Open-ended question answering with charts," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 11 817–11 837.

[26] M. Huang, H. Lai, X. Zhang, W. Wu, J. Ma, L. Zhang, and J. Liu, "EvoChart: A benchmark and a self-training approach towards real-world chart understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 4, 2025, pp. 3680–3688.

[27] J. Obeid and E. Hoque, "Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model," *arXiv preprint arXiv:2010.09142*, 2020.

[28] S. Kantharaj, R. T. Leong, X. Lin, A. Masry, M. Thakkar, E. Hoque, and S. Joty, "Chart-to-Text: A large-scale benchmark for chart summarization," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2022, pp. 4005–4023.

[29] A. Masry, M. Shahmohammadi, M. R. Parvez, E. Hoque, and S. Joty, "ChartInstruct: Instruction tuning for chart comprehension and reasoning," in *Findings of the Association for Computational Linguistics*, 2024, pp. 10 387–10 409.

[30] C. Wu, Y. Ge, Q. Guo, J. Wang, Z. Liang, Z. Lu, Y. Shan, and P. Luo, "Plot2Code: A comprehensive benchmark for evaluating multi-modal large language models in code generation from scientific plots," *arXiv preprint arXiv:2405.07990*, 2024.

[31] Z. Yang, Z. Zhou, S. Wang, X. Cong, X. Han, Y. Yan, Z. Liu, Z. Tan, P. Liu, D. Yu, Z. Liu, X. Shi, and M. Sun, "MatPlotAgent: Method and evaluation for LLM-based agentic scientific data visualization," in *Findings of the Association for Computational Linguistics*, 2024, pp. 11 789–11 804.

[32] Z. Wang, M. Xia, L. He, H. Chen, Y. Liu, R. Zhu, K. Liang, X. Wu, H. Liu, S. Malladi *et al.*, "CharXiv: Charting gaps in realistic chart understanding in multimodal LLMs," *Advances in Neural Information Processing Systems*, vol. 37, pp. 113 569–113 697, 2024.

[33] T.-Y. Hsu, C. L. Giles, and T.-H. Huang, "SciCap: Generating captions for scientific figures," in *Findings of the Association for Computational Linguistics: EMNLP*, 2021, pp. 3258–3264.

[34] Z. Zhang, W. Ma, and S. Vosoughi, "Is GPT-4V (ision) all you need for automating academic data visualization? exploring vision-language models' capability in reproducing academic charts," in *Findings of the Association for Computational Linguistics: EMNLP*, 2024, pp. 8271–8288.

[35] F. Liu, J. Eisenschlos, F. Piccinno, S. Krichene, C. Pang, K. Lee, M. Joshi, W. Chen, N. Collier, and Y. Altun, "DePlot: One-shot visual language reasoning by plot-to-table translation," in *Findings of the Association for Computational Linguistics*, 2023, pp. 10 381–10 399.

[36] F. Liu, F. Piccinno, S. Krichene, C. Pang, K. Lee, M. Joshi, Y. Altun, N. Collier, and J. Eisenschlos, "MatCha: Enhancing visual language pretraining with math reasoning and chart derendering," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 12 756–12 770.

[37] A. Masry, P. Kavehzadeh, X. L. Do, E. Hoque, and S. Joty, "UniChart: A universal vision-language pretrained model for chart comprehension and reasoning," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 14 662–14 684.

[38] F. Meng, W. Shao, Q. Lu, P. Gao, K. Zhang, Y. Qiao, and P. Luo, "ChartAssistant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning," in *Findings of the Association for Computational Linguistics*, 2024, pp. 7775–7803.

[39] C. Yang, C. Shi, Y. Liu, B. Shui, J. Wang, M. Jing, L. XU, X. Zhu, S. Li, Y. Zhang, G. Liu, X. Nie, D. Cai, and Y. Yang, "ChartMimic: Evaluating LMM's cross-modal reasoning capability via chart-to-code generation," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=sGpCzsfd1K

[40] A. Masry, M. S. Islam, M. Ahmed, A. Bajaj, F. Kabir, A. Kartha, M. T. R. Laskar, M. Rahman, S. Rahman, M. Shahmohammadi *et al.*, "ChartQAPro: A more diverse and challenging benchmark for chart question answering," *arXiv preprint arXiv:2504.05506*, 2025.

[41] M. Mathew, V. Bagal, R. Tito, D. Karatzas, E. Valveny, and C. Jawahar, "InfographicVQA," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1697–1706.

[42] "Image deduplicator (imagededup)," 2019, accessed: 2025-05-07. [Online]. Available: https://github.com/idealo/imagededup

[43] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the International Conference on Machine Learning*, 2021, pp. 8748–8763.

[44] "Data Viz Project," 2025, accessed: 2025-05-07. [Online]. Available: https://datavizproject.com/

[45] "Datylon," 2025, accessed: 2025-05-07. [Online]. Available: https://www.datylon.com/blog/types-of-charts-graphs-examples-data-visualization

[46] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.

[47] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, X. Wang, and Y. Qiao, "InternImage: Exploring large-scale vision foundation models with deformable convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 408–14 419.

[48] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. Ni, and H.-Y. Shum, "DINO: DETR with improved denoising anchor boxes for end-to-end object detection," in *The International Conference on Learning Representations*, 2023.

[49] "PaddleOCR-PPOCR-v4," 2025, accessed: 2025-05-07. [Online]. Available: https://github.com/PaddlePaddle/PaddleOCR

[50] M. Otani, N. Inoue, K. Kikuchi, and R. Togashi, "LTSim: Layout transportation-based similarity measure for evaluating layout generation," *arXiv preprint arXiv:2407.12356*, 2024.

[51] K. Hu, S. Gaikwad, M. Hulsebos, M. A. Bakker, E. Zgraggen, C. Hidalgo, T. Kraska, G. Li, A. Satyanarayan, and Ç. Demiralp, "VizNet: Towards a large-scale visualization learning and benchmarking repository," in *Proceedings of the Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[52] "UNdata," 2025, accessed: 2025-05-07. [Online]. Available: https://data.un.org/datamartinfo.aspx

[53] OWID, "Our World in Data," https://ourworldindata.org/, 2025, accessed: 2025-04-30.

[54] Papers with Code, "Machine Learning Datasets," https://paperswithcode.com/datasets, 2025, accessed: 2025-04-30.

[55] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang, "DataShot: Automatic generation of fact sheets from tabular data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 895–905, 2020.

[56] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, 2019, pp. 3982–3992.

[57] P. Lu, L. Qiu, J. Chen, T. Xia, Y. Zhao, W. Zhang, Z. Yu, X. Liang, and S.-C. Zhu, "IconQA: A new benchmark for abstract diagram understanding and visual language reasoning," *arXiv preprint arXiv:2110.13214*, 2021.

[58] "Flaticon," 2025, accessed: 2025-05-07. [Online]. Available: https://www.flaticon.com/

[59] "Noun project," 2025, accessed: 2025-05-07. [Online]. Available: https://thenounproject.com/

[60] Y. Luo, X. Qin, N. Tang, and G. Li, "DeepEye: Towards automatic data visualization," in *Proceedings of the IEEE International Conference on Data Engineering*, 2018, pp. 101–112.

[61] S. Liu, M. Tao, Y. Huang, C. Wang, and C. Li, "Image-driven harmonious color palette generation for diverse information visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 7, pp. 3089–3103, 2024.

[62] J. Chen, W. Yang, Z. Jia, L. Xiao, and S. Liu, "Dynamic color assignment for hierarchical data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 1, pp. 338–348, 2025.

[63] C. K. Coursaris and K. Kripintris, "Web aesthetics and usability: An empirical study of the effects of white space," *International Journal of E-Business Research*, vol. 8, no. 1, pp. 35–53, 2012.

[64] A. Lodi, S. Martello, and D. Vigo, "Approximation algorithms for the oriented two-dimensional bin packing problem," *European Journal of Operational Research*, vol. 112, no. 1, pp. 158–166, 1999.

[65] J. Zhu, W. Wang, Z. Chen, Z. Liu, S. Ye, L. Gu, Y. Duan, H. Tian, W. Su, J. Shao *et al.*, "InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models," *arXiv preprint arXiv:2504.10479*, 2025.

[66] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang *et al.*, "Qwen2.5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.

[67] C. Si, Y. Zhang, R. Li, Z. Yang, R. Liu, and D. Yang, "Design2Code: Benchmarking multimodal code generation for automated front-end engineering," in *Proceedings of the Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2025, pp. 3956–3974.

[68] D. F. Crouse, "On implementing 2D rectangular assignment algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1679–1696, 2016.

[69] C. Chen, Y. Guo, F. Tian, S. Liu, W. Yang, Z. Wang, J. Wu, H. Su, H. Pfister, and S. Liu, "A unified interactive model evaluation for classification, object detection, and instance segmentation in computer vision," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 1, pp. 76–86, 2024.

[70] "Gemini 2.5 pro preview: even better coding performance," 2025, accessed: 2025-05-07. [Online]. Available: https://developers.googleblog.com/en/gemini-2-5-pro-io-improved-coding-performance/

[71] "Introducing GPT-4.1 in the API," 2025, accessed: 2025-05-07. [Online]. Available: https://openai.com/index/gpt-4-1/

[72] "Claude 3.7 Sonnet and Claude Code," 2025, accessed: 2025-05-07. [Online]. Available: https://www.anthropic.com/news/claude-3-7-sonnet

[73] "Introducing OpenAI o3 and o4-mini," 2025, accessed: 2025-05-07. [Online]. Available: https://openai.com/index/introducing-o3-and-o4-mini/

[74] "Start building with Gemini 2.5 Flash," 2025, accessed: 2025-05-07. [Online]. Available: https://developers.googleblog.com/en/start-building-with-gemini-25-flash/

[75] "Introducing OpenAI o1," 2025, accessed: 2025-05-07. [Online]. Available: https://openai.com/o1/

[76] "Hello GPT-4o," 2025, accessed: 2025-05-07. [Online]. Available: https://openai.com/index/hello-gpt-4o/

[77] "Doubao-1.5-vision-pro," 2025, accessed: 2025-05-07. [Online]. Available: https://www.volcengine.com/docs/82379/1553586

[78] "Moonshot AI," 2025, accessed: 2025-05-07. [Online]. Available: https://platform.moonshot.cn/docs/intro

[79] "The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation," 2025, accessed: 2025-05-07. [Online]. Available: https://ai.meta.com/blog/llama-4-multimodal-intelligence/

[80] "Introducing our latest image generation model in the API," 2025, accessed: 2025-05-07. [Online]. Available: https://openai.com/index/image-generation-api/

[81] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.

[82] M. R. Luo, G. Cui, and B. Rigg, "The development of the cie 2000 colour-difference formula: Ciede2000," *Color Research & Application*, vol. 26, no. 5, pp. 340–350, 2001.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: In this paper, we build a million-scale dataset consisting of 1,151,087 programmatically created infographic charts and 104,519 real infographic charts, and showcase its utility through three tasks. The method for dataset construction is introduced in Sec. 3, and experiments for the three tasks are presented in Sec. 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitation is discussed in Sec. 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [NA]

Justification: This paper does not include any theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide the detailed experimental settings in Supp. D, and release the data and code used in the experiments through the links provided after the abstract.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

Answer: [Yes]

Justification: The data and code are available through the links included after the abstract.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide the detailed experimental settings in Supp. D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report confidence intervals and Wilcoxon signed-rank test statistics for the evaluation results in Sec. 4.3. Due to resource constraints, variance is not reported for other experiments, as they depend on GPU resources and paid APIs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide the computing resources needed for the experiments in Supp. D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics and discuss the ethical considerations in Supp. F.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the societal impacts of the paper in Supp. F.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [Yes]

    Justification: We apply large multimodal models to automatically filter out-of-scope scraped data, as detailed in Sec. 3.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We provide links to the main websites and their terms of service for all online platforms from which data was collected, as detailed in Supp. A.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We host our data and code on GitHub and Hugging Face, accompanied by comprehensive documentation. Links are provided after the abstract.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [Yes]

    Justification: We provide the details of the user study (Sec. 4.3) in Supp. D.3, including instructions, interface screenshots, and compensation.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [Yes]

    Justification: For the user study in Sec. 4.3, we do not see potential risks and have received IRB approval.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: We use LLM for data processing, as introduced in Sec. 3.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Chart-rich Websites

Table 5: Chart-rich websites and their terms of service.

Name	URL	Terms of Service
Pinterest	pinterest.com	Link[1]
Statista	statista.com	Link[2]
Visual Capitalist	visualcapitalist.com	Link[3]
World Statistics	world-statistics.org	Link[4]
Chart Porn	chartporn.org	Link[5]
Cool Infographics	coolinfographics.com/blog	Link[6]
Daily Infographics	dailyinfographics.eu	Link[7]
Data Pointed	datapointed.net	Link[8]
DataRep	datarep.com	Link[9]
Flickr	flickr.com	Link[10]
FlowingData	flowingdata.com	Link[11]
Infographics Archive	infographicsarchive.com	Link[12]
Information is Beautiful	informationisbeautiful.net	Link[13]
National Post	nationalpost.com	Link[14]
Pikto Chart	piktochart.com	Link[15]
SCMP	scmp.com	Link[16]
Visme	visme.co	Link[17]
Visual Cinnamon	visualcinnamon.com	Link[18]
Chit Chart	chitchart.com	Link[19]

[1] https://policy.pinterest.com/en/terms-of-service
[2] https://www.statista.com/getting-started/publishing-statista-content
[3] https://licensing.visualcapitalist.com/recent-changes-to-visual-capitalist-licensing/
[4] https://world-statistics.org/terms-of-use.php
[5] https://chartporn.org/about/
[6] https://coolinfographics.com/about
[7] https://dailyinfographic.com/terms
[8] http://www.datapointed.net/about/
[9] https://www.datarep.com/website-terms-of-use/
[10] https://www.flickr.com/help/terms
[11] https://flowingdata.com/about/
[12] https://www.infographicsarchive.com/about/
[13] https://informationisbeautiful.net/about
[14] https://www.postmedia.com/terms-and-conditions
[15] https://piktochart.com/terms-of-use/
[16] https://www.scmp.com/terms-conditions
[17] https://www.visme.co/about/
[18] https://www.visualcinnamon.com/about/
[19] https://chitchart.com/terms-use

B Chart Types and Variations

We provide a full list of 75 chart types and 330 chart variations.

B.1 Chart Types

We summarize chart types observed in the collected real infographic charts, drawing on two existing taxonomies: Data Viz Project [44] and Datylon [45]. The diversity of chart types ensures that our synthetic infographic charts can adapt to a wider range of scenarios and data representations, making them valuable for model training and evaluation. The full list of 75 chart types is in Figs. 5-20.

B.2 Chart Variations

For each chart type, we extract chart variations from real-world infographic charts, featuring diverse visual styles, *e.g.*, element shapes, icon placement, hand-drawn styles, and 3D styles. Therefore, we provide illustrations of the variations under each chart type (Figs. 5-20).

B.3 Mapping Rules

We determine chart types by analyzing the data attributes and their characteristics. First, we apply rule-based mapping (Table 6) that identifies candidate chart types based on data attribute combinations. If multiple candidates remain, we instruct Gemini-2.0-Flash to select the best fit by evaluating the compatibility between the data and these chart types. The selection prompt, shown below, includes candidate chart types, their descriptions, and key data statistics to guide the decision.

Table 6: Mapping rules defining attribute combinations for chart type selection. C, N, and T are abbreviations for Categorical, Numeric, and Temporal attributes, respectively. The notation $X \times k$ indicates k distinct attributes of type X. When a symbol such as $*$ is specified (e.g., for the Diverging Bar Chart), it indicates that the first categorical attribute must contain exactly two distinct values.

Chart Type	**Attribute Combinations**
Vertical Bar Chart	C $\times$ 1 + N $\times$ 1
Vertical Stacked Bar Chart	C $\times$ 2 + N $\times$ 1
Vertical Grouped Bar Chart	C $\times$ 2 + N $\times$ 1
Horizontal Bar Chart	C $\times$ 1 + N $\times$ 1
Horizontal Stacked Bar Chart	C $\times$ 2 + N $\times$ 1
Horizontal Grouped Bar Chart	C $\times$ 2 + N $\times$ 1
Radial Bar Chart	C $\times$ 1 + N $\times$ 1
Radial Stacked Bar Chart	C $\times$ 2 + N $\times$ 1
Radial Grouped Bar Chart	C $\times$ 2 + N $\times$ 1
Circular Bar Chart	C $\times$ 1 + N $\times$ 1
Circular Stacked Bar Chart	C $\times$ 2 + N $\times$ 1
Circular Grouped Bar Chart	C $\times$ 2 + N $\times$ 1
Pictorial Percentage Bar Chart	C $\times$ 1 + N $\times$ 1
Histogram	C $\times$ 1 + N $\times$ 1, T $\times$ 1 + N $\times$ 1
Lollipop Chart	C $\times$ 1 + N $\times$ 1
Dot chart	C $\times$ 1 + N $\times$ 1, C $\times$ 2 + N $\times$ 1
Diverging Bar Chart	C $\times$ 2 + N $\times$ 1 $*$
Vertical Bar Chart With Circle	C $\times$ 1 + N $\times$ 2
Horizontal Bar Chart With Circle	C $\times$ 1 + N $\times$ 2
Vertical Dot Bar Chart	C $\times$ 1 + N $\times$ 1
Horizontal Dot Bar Chart	C $\times$ 1 + N $\times$ 1
Dumbbell Plot	T $\times$ 1 + N $\times$ 1 + C $\times$ 1 $*$
Span Chart	C $\times$ 1 + N $\times$ 2
Bump Chart	T $\times$ 1 + N $\times$ 1 + C $\times$ 1
Line Graph	T $\times$ 1 + N $\times$ 1, T $\times$ 1 + N $\times$ 1 + C $\times$ 1
Spline Graph	T $\times$ 1 + N $\times$ 1, T $\times$ 1 + N $\times$ 1 + C $\times$ 1

(Continued on next page)

Table 6: (Continued): Mapping rules defining attribute combinations for chart type selection.

Chart Type	**Attribute Combinations**
Stepped Line Graph	$T \times 1 + N \times 1$, $T \times 1 + N \times 1 + C \times 1$
Slope Chart	$T \times 1 + N \times 1 + C \times 1$
Small Multiples of Line Graphs	$T \times 1 + N \times 1 + C \times 1$
Small Multiples of Spline Graphs	$T \times 1 + N \times 1 + C \times 1$
Small Multiples of Stepped Line Graphs	$T \times 1 + N \times 1 + C \times 1$
Area Chart	$T \times 1 + N \times 1$, $T \times 1 + N \times 1 + C \times 1$
Spline Area Chart	$T \times 1 + N \times 1$, $T \times 1 + N \times 1 + C \times 1$
Layered Area Chart	$T \times 1 + N \times 1 + C \times 1$
Layered Spline Area Chart	$T \times 1 + N \times 1 + C \times 1$
Range Area Chart	$T \times 1 + N \times 1 + C \times 1$ *
Stacked Area Chart	$T \times 1 + N \times 1 + C \times 1$
Radial Area Chart	$T \times 1 + N \times 1 + C \times 1$
Radial Spline Area Chart	$T \times 1 + N \times 1 + C \times 1$
Radial Layered Area Chart	$T \times 1 + N \times 1 + C \times 1$
Radial Layered Spline Area Chart	$T \times 1 + N \times 1 + C \times 1$
Radial Range Area Chart	$T \times 1 + N \times 1 + C \times 1$ *
Radial Stacked Area Chart	$T \times 1 + N \times 1 + C \times 1$
Diverging Area Chart	$T \times 1 + N \times 1 + C \times 1$ *
Diverging Spline Area Chart	$T \times 1 + N \times 1 + C \times 1$ *
Small Multiples of Area Charts	$T \times 1 + N \times 1 + C \times 1$
Small Multiples of Spline Area Charts	$T \times 1 + N \times 1 + C \times 1$
Pie Chart	$C \times 1 + N \times 1$
Donut Chart	$C \times 1 + N \times 1$
Semicircle Pie Chart	$C \times 1 + N \times 1$
Semicircle Donut Chart	$C \times 1 + N \times 1$
Rose Chart	$C \times 1 + N \times 1$
Small Multiples of Pie Charts	$C \times 2 + N \times 1$
Small Multiples of Donut Charts	$C \times 2 + N \times 1$
Small Multiples of Semicircle Pie Charts	$C \times 2 + N \times 1$
Small Multiples of Semicircle Donut Charts	$C \times 2 + N \times 1$
Small Multiples of Rose Charts	$C \times 2 + N \times 1$
Radar Line Chart	$C \times 1 + N \times 1$
Radar Spline Chart	$C \times 1 + N \times 1$
Small Multiples of Radar Line Charts	$C \times 2 + N \times 1$
Small Multiples of Radar Spline Charts	$C \times 2 + N \times 1$
Proportional Area Chart	$C \times 1 + N \times 1$
Scatterplot	$C \times 1 + N \times 2$
Grouped Scatterplot	$C \times 2 + N \times 2$
Bubble Chart	$C \times 1 + N \times 2$
Heatmap	$N \times 2$
Waffle Chart	$N \times 1$
Small Multiples of Waffle Charts	$C \times 1 + N \times 1$
Alluvial Diagram	$C \times 1 + N \times 1 + T \times 1$, $C \times 2 + N \times 1$
Gauge Chart	$N \times 1$
Small Multiples of Gauge Charts	$C \times 1 + N \times 1$
Funnel Chart	$C \times 1 + N \times 1$
Pyramid Chart	$C \times 1 + N \times 1$
Treemap	$C \times 2 + N \times 1$
Voronoi Treemap	$C \times 2 + N \times 1$

 # INPUT
Candidate Chart Types: `{Candidate Chart Types}`
Chart Descriptions: `{Chart Descriptions}`
Attribute Statistics: `{Attribute Statistics}`

Keep answers concise and direct.

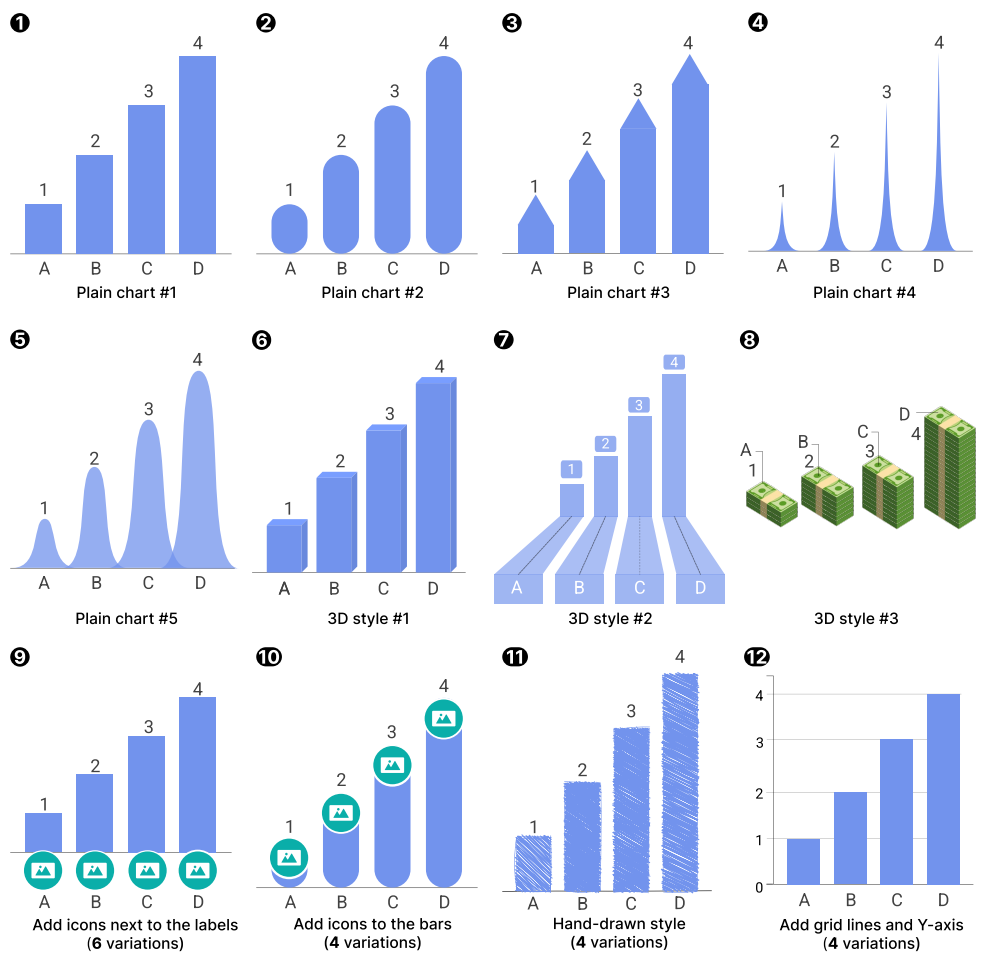
 # INSTRUCTION
Role: You are an expert assistant in data visualization and chart selection.
Task: Select the *single most optimal* chart type from `{Candidate Chart Types}` using their `{Chart Descriptions}` and `{Attribute Statistics}`. Prioritize **data-chart compatibility**: the chart must clearly and accurately represent key data insights.
Instructions for Selection:

1. Analyze all provided inputs.
2. Leverage your expertise to interpret how `{Attribute Statistics}` (e.g., categorical cardinality, number of temporal points, numerical value ranges, cumulative nature of data) impact the effectiveness and clarity of each candidate chart type, considering their `{Chart Descriptions}`.
3. Evaluate chart types based on descriptions, common visualization best practices, and your interpretation of `{Attribute Statistics}` to identify the most insightful and unambiguous visualization.
4. Primary goal: maximize data-chart compatibility.

Output Format: Return *only* the name of the single selected chart type.

 # EXAMPLE OF TASK EXECUTION
Input:
Candidate Chart Types: `["Line Graph", "Area Chart", "Spline Graph"]`
Chart Descriptions: `"Line Graph: Emphasizes trends and rate of change over time; best for non-cumulative data with sufficient points. Area Chart: Shows trends and volume/magnitude over time; good for cumulative data or showing part-to-whole over time. Spline Graph: A Line Graph with smoothed curves, visually softens trends, suitable for data with many points or where a less angular look is desired."`
Attribute Statistics: `{ "temporal_points": 30, "numeric_min_value": 15, "numeric_max_value": 450 }`

Output:
`Line Graph`



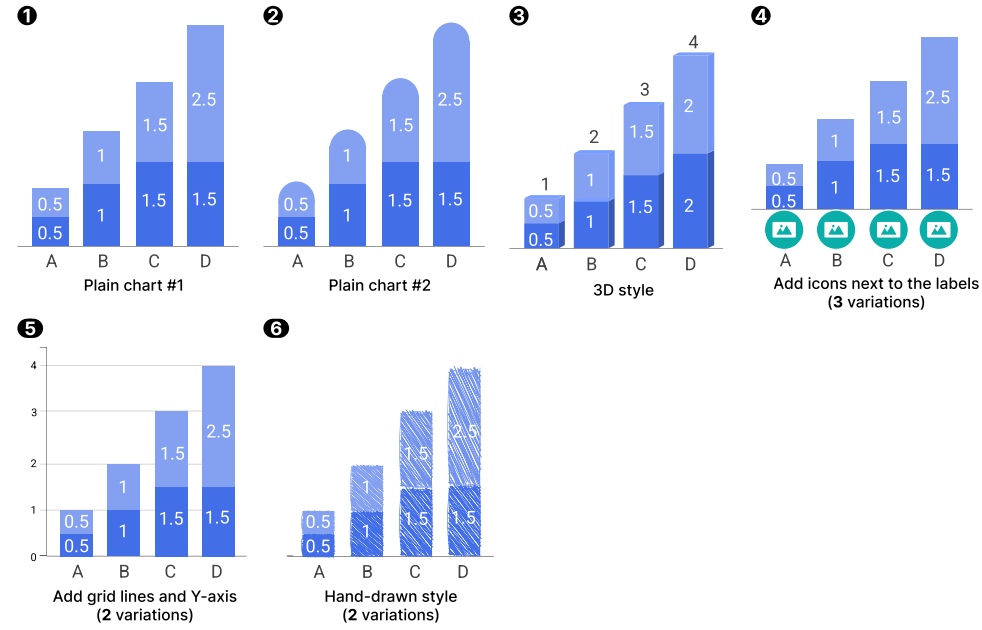


Figure 5: 75 chart types and 330 chart variations (Part 1).

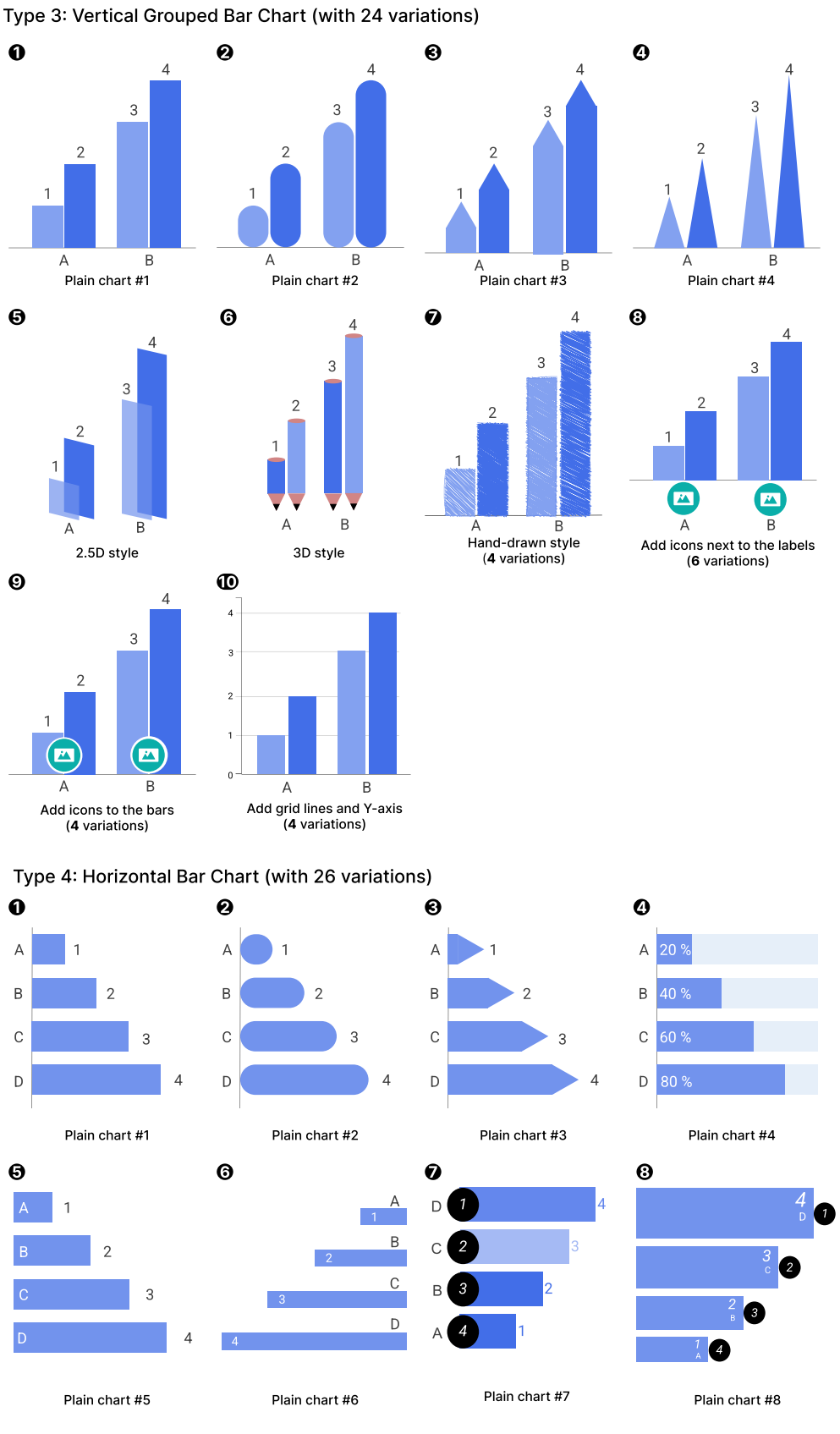
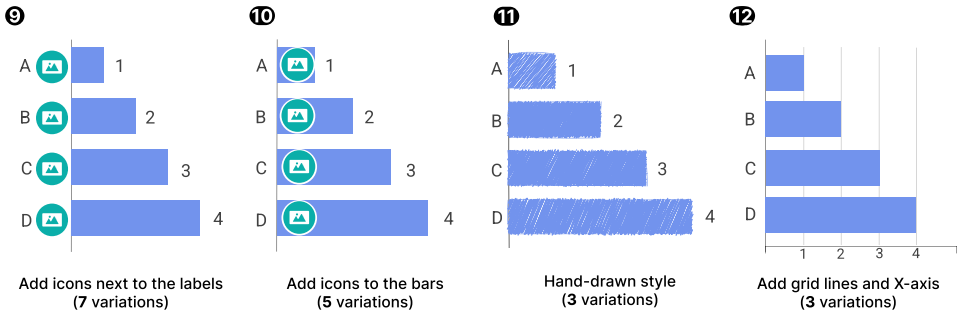
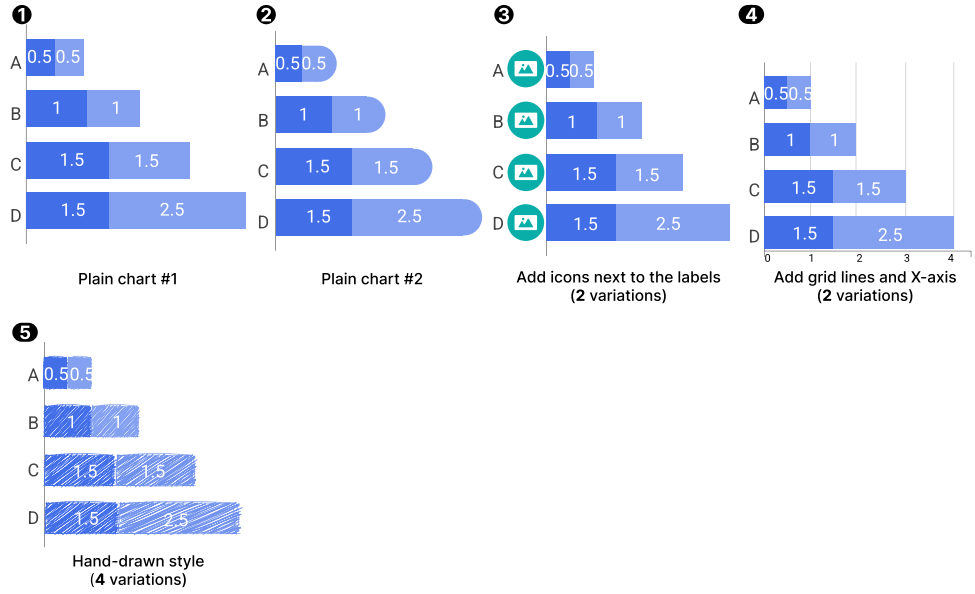


Figure 6: 75 chart types and 330 chart variations (Part 2).



Type 5: Horizontal Stacked Bar Chart (with 10 variations)



Type 6: Horizontal Grouped Bar Chart (with 14 variations)

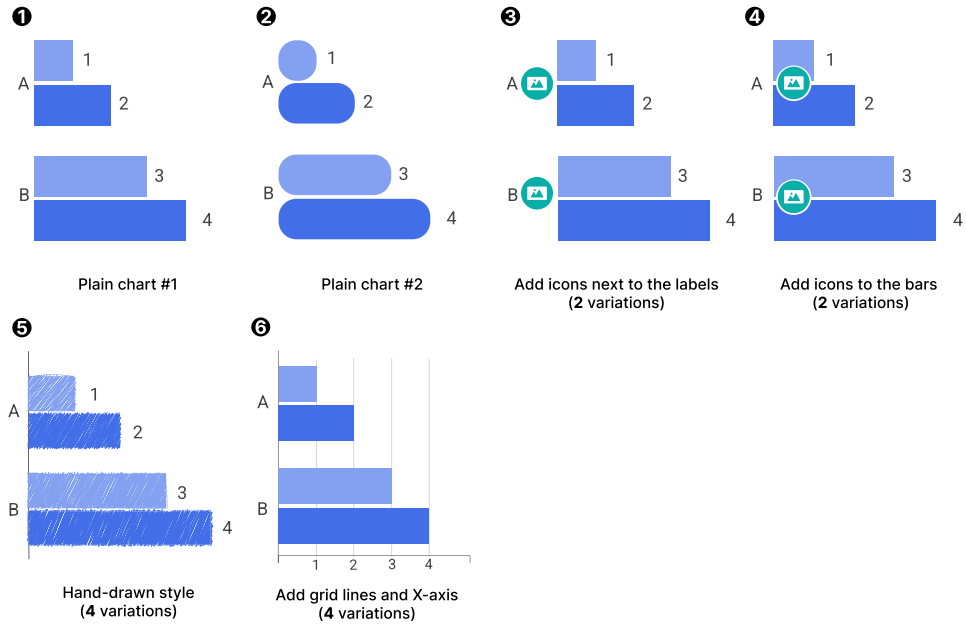


Figure 7: 75 chart types and 330 chart variations (Part 3).

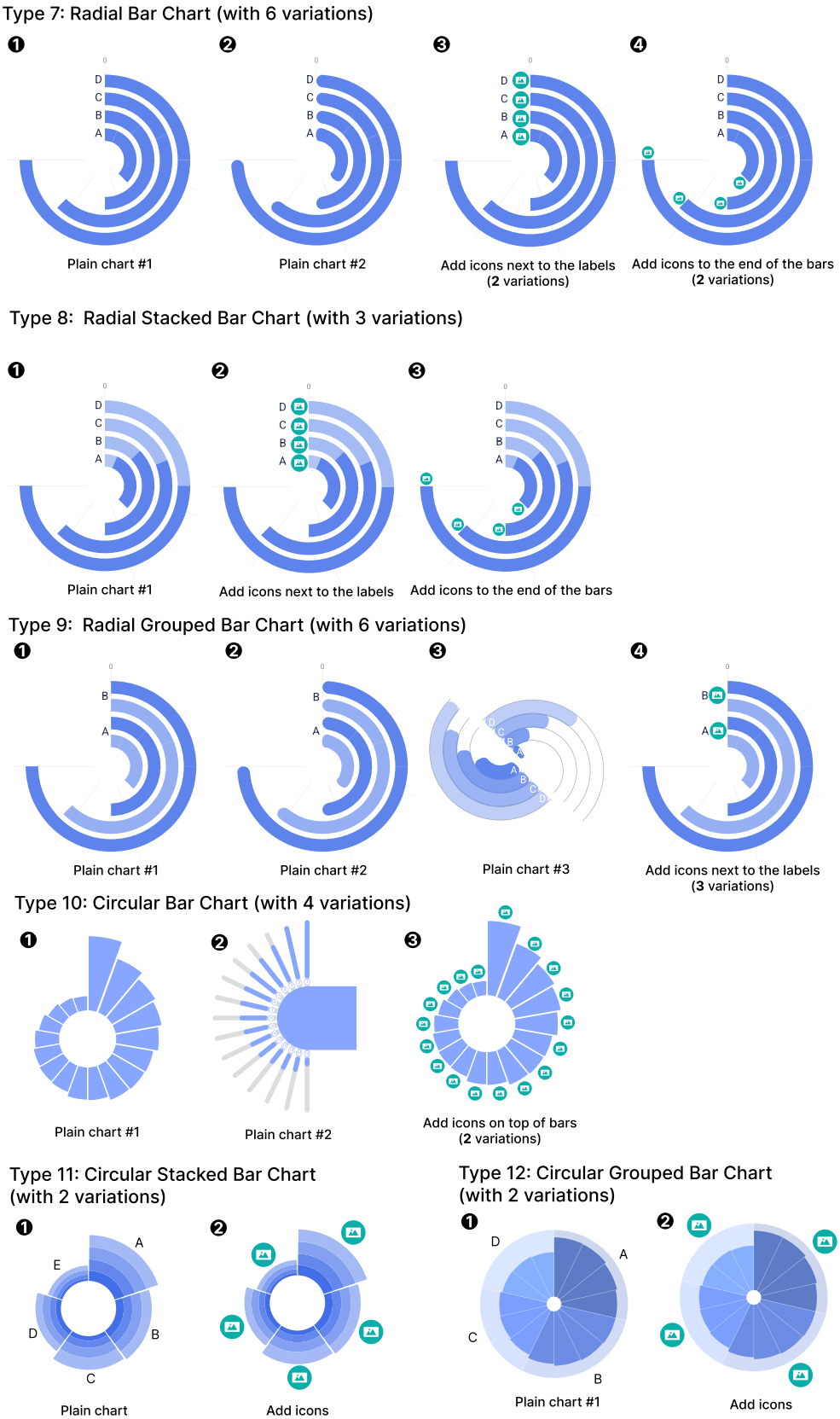
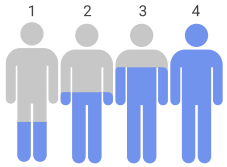


Figure 8: 75 chart types and 330 chart variations (Part 4).

Type 13: Pictorial Percentage Bar Chart (with 1 variation)

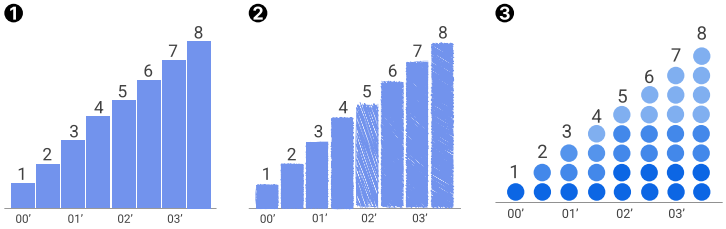
❶



Plain chart #1

Type 14: Histogram (with 3 variations)

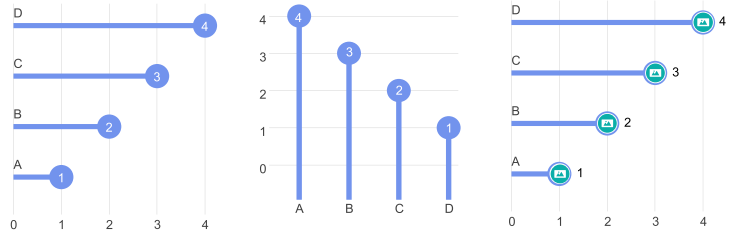
❶ ❷ ❸



Plain chart #1 Hand-drawn style Replace bar with circles

Type 15: Lollipop Chart (with 4 variations)

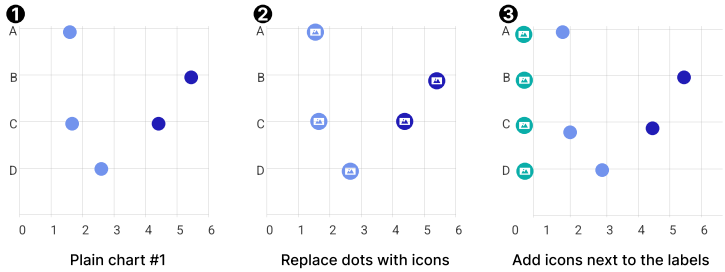
❶ ❷ ❸



Plain chart #1 Plain chart #2 Add icons to lollipops (2 variations)

Type 16: Dot Chart (with 3 variations)

❶ ❷ ❸



Plain chart #1 Replace dots with icons Add icons next to the labels

Figure 9: 75 chart types and 330 chart variations (Part 5).

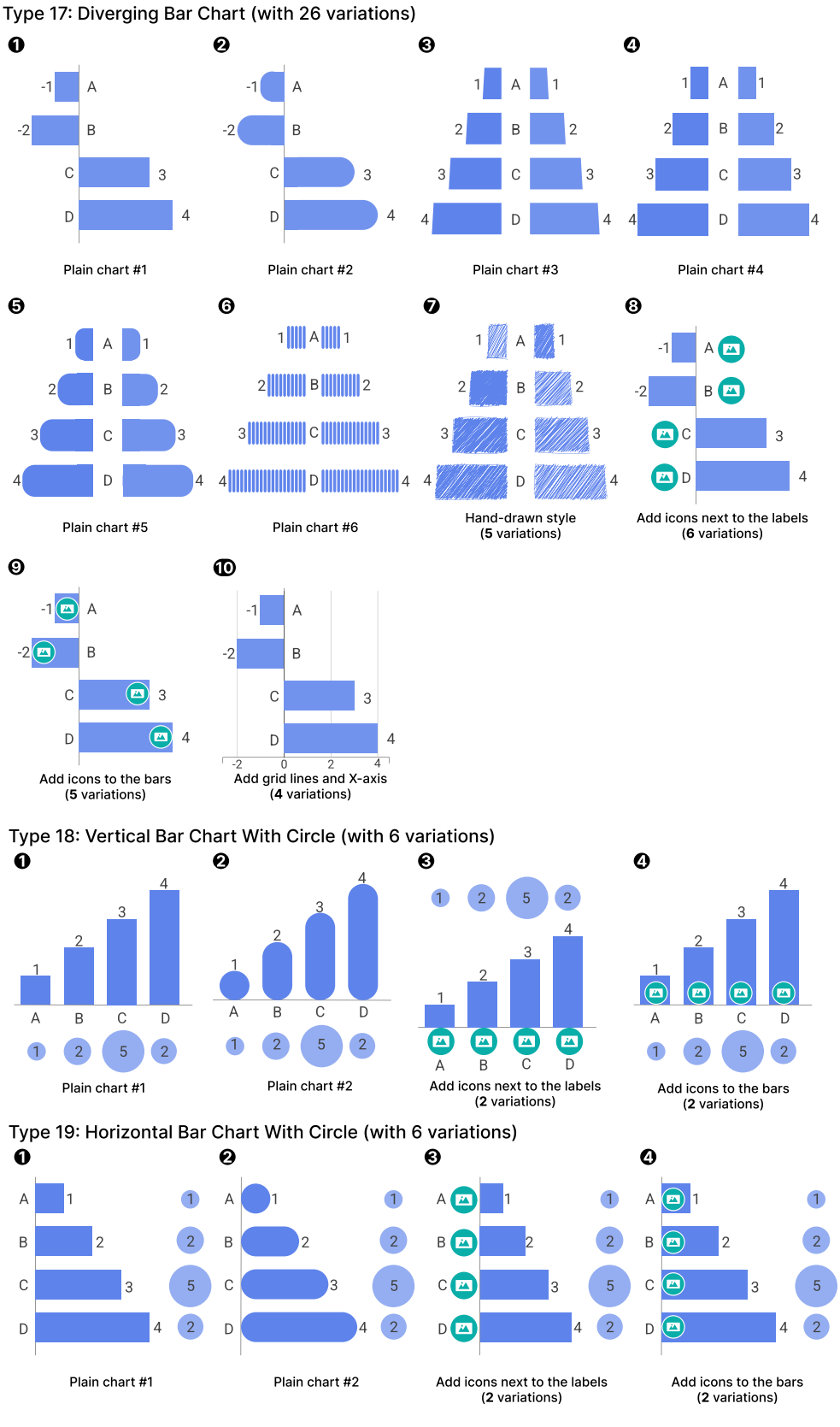


Figure 10: 75 chart types and 330 chart variations (Part 6).

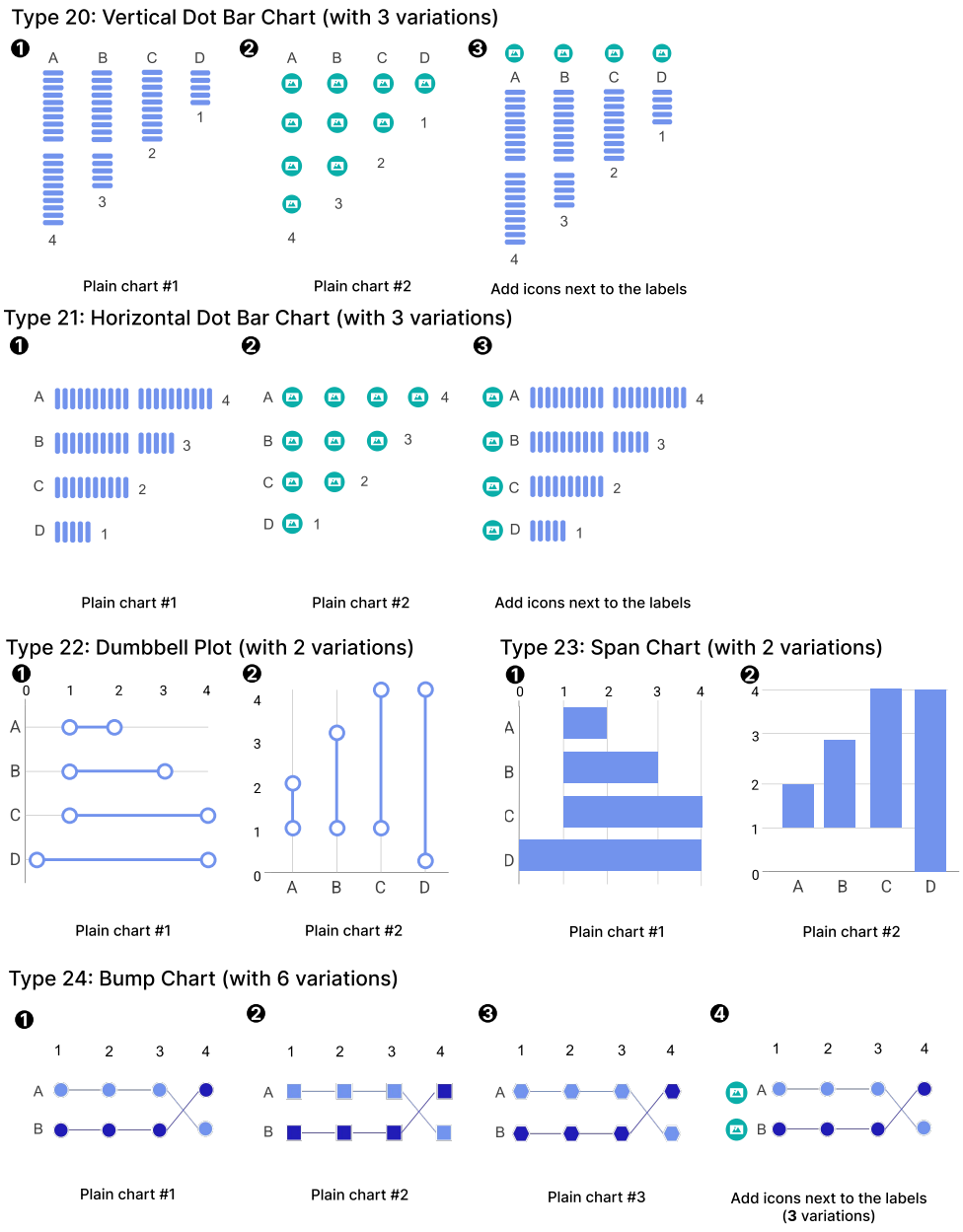
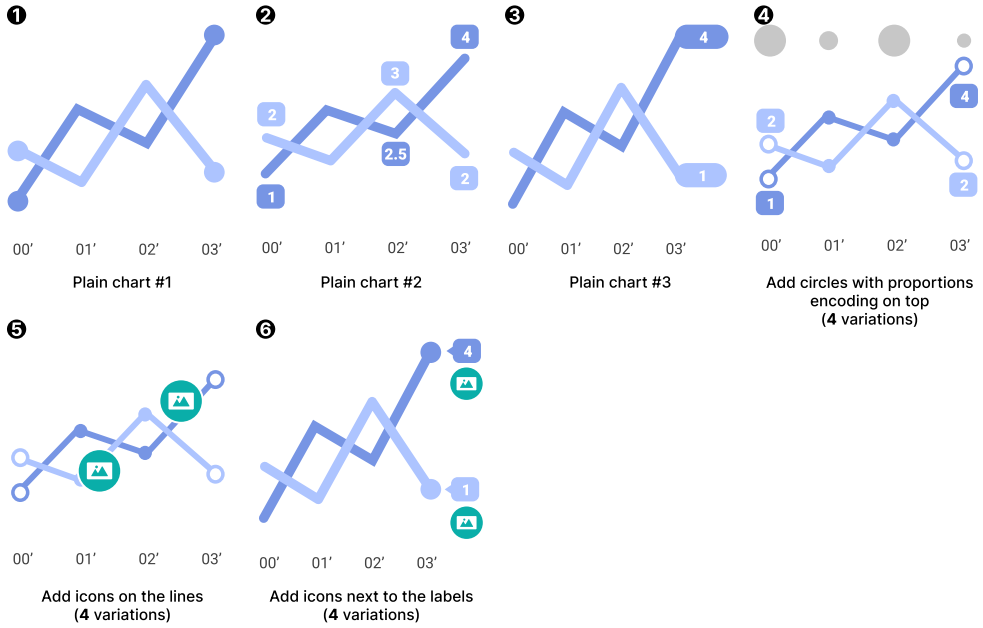
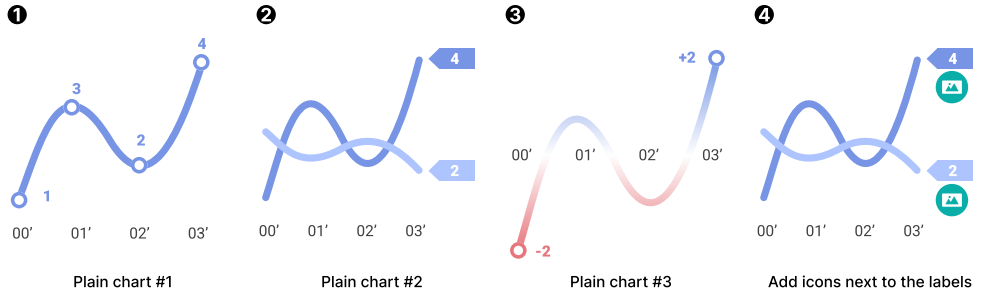


Figure 11: 75 chart types and 330 chart variations (Part 7).

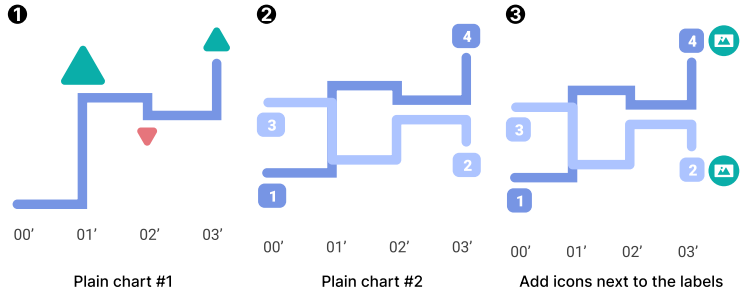
Type 25: Line Graph (with 15 variations)



Type 26: Spline Graph (with 4 variations)



Type 27: Stepped Line Graph (with 3 variations)



Type 28: Slope Chart (with 2 variations)

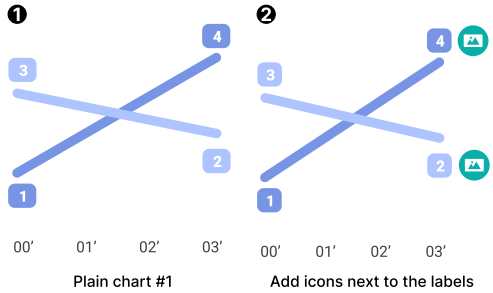


Figure 12: 75 chart types and 330 chart variations (Part 8).

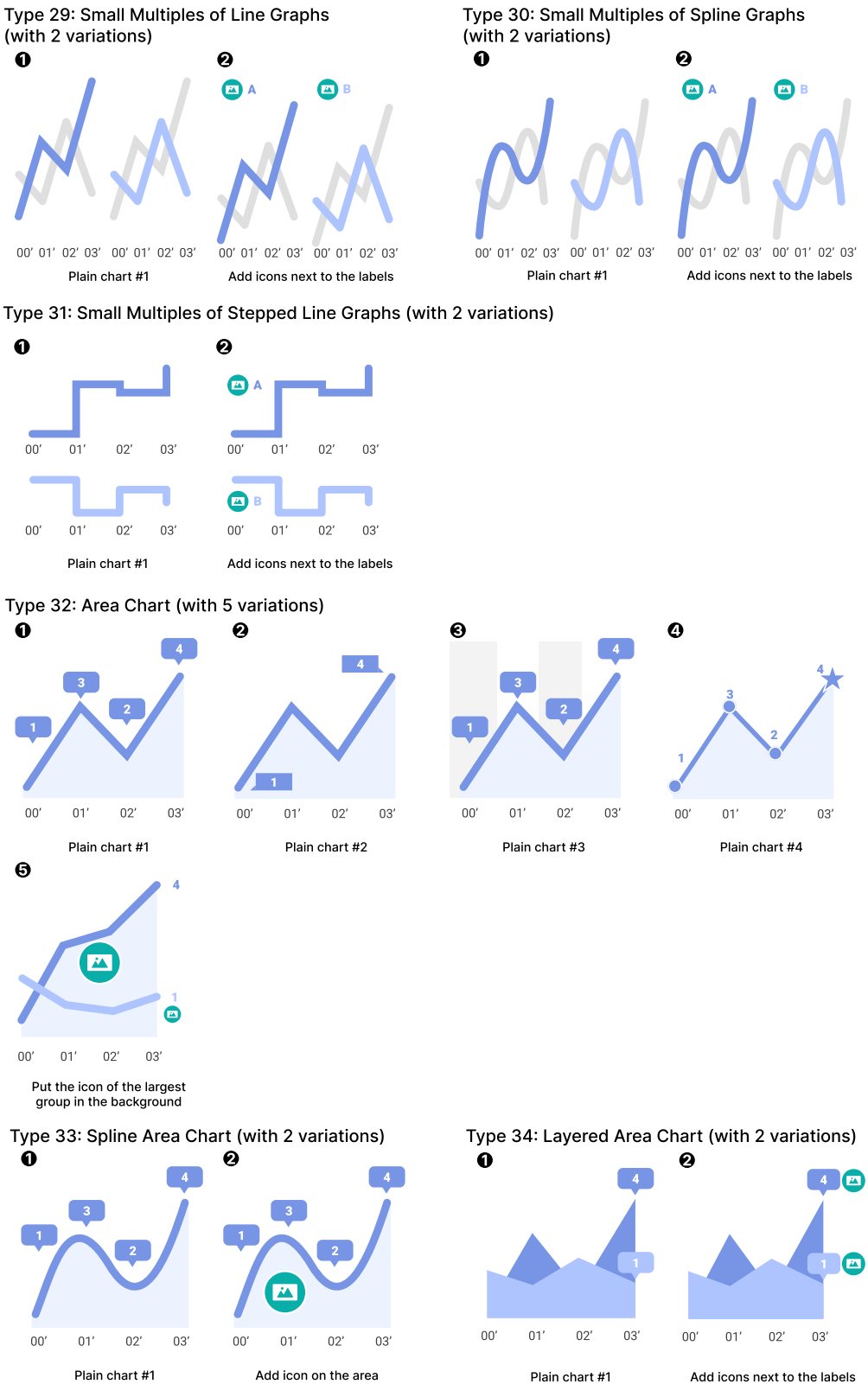


Figure 13: 75 chart types and 330 chart variations (Part 9).

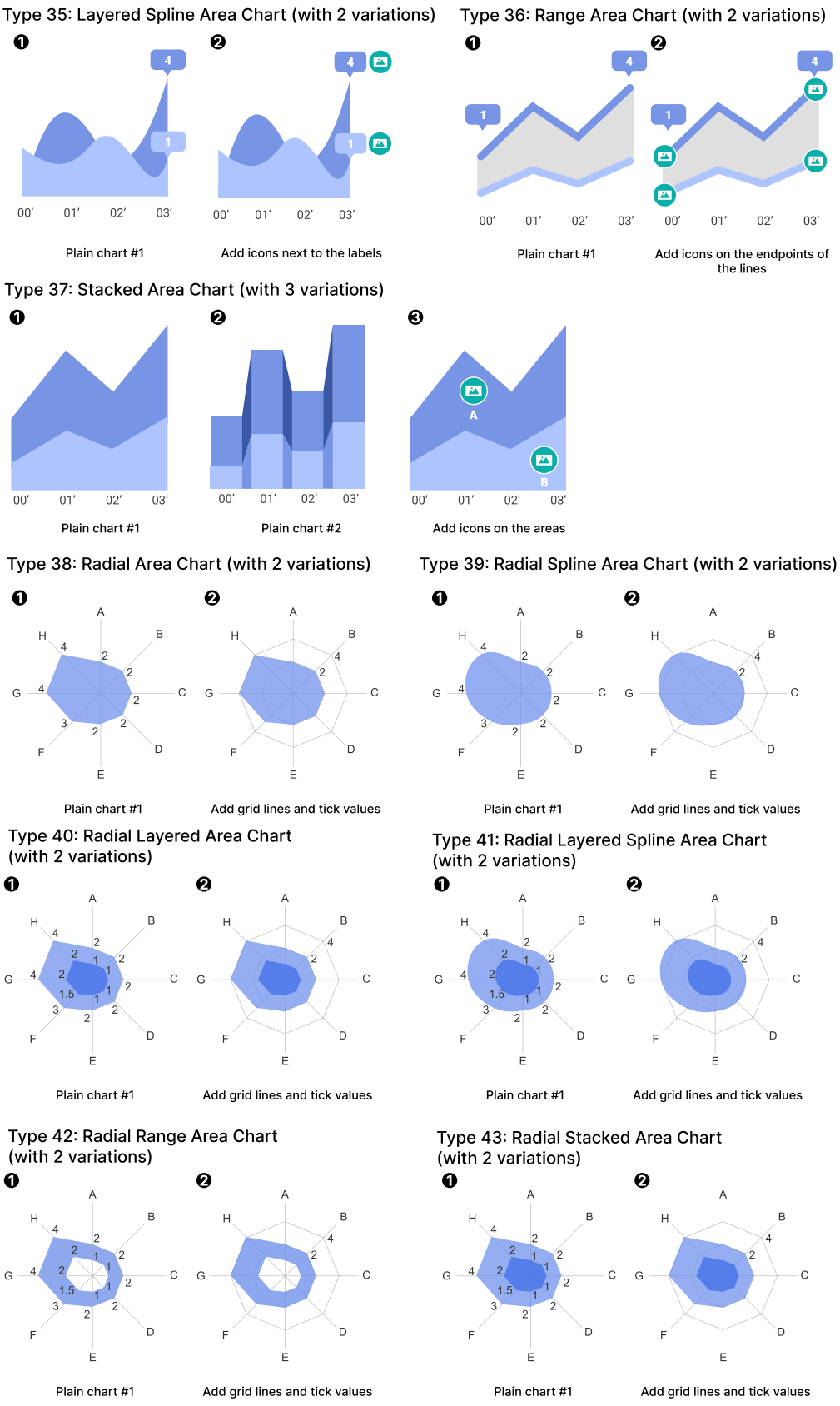
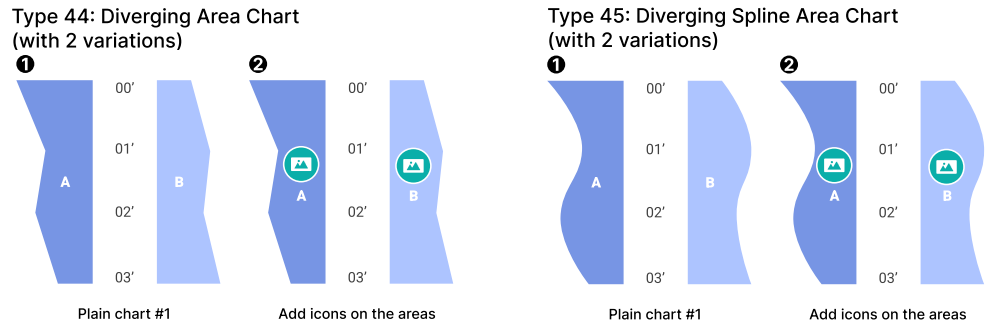
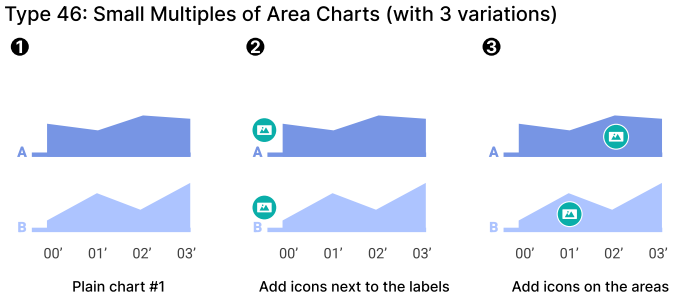
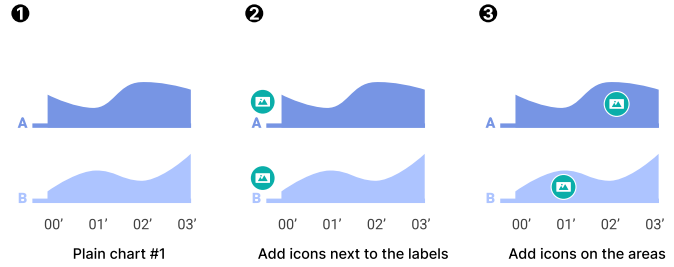
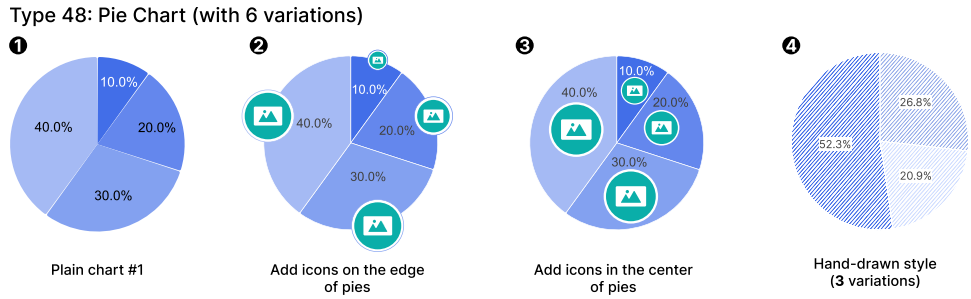


Figure 14: 75 chart types and 330 chart variations (Part 10).









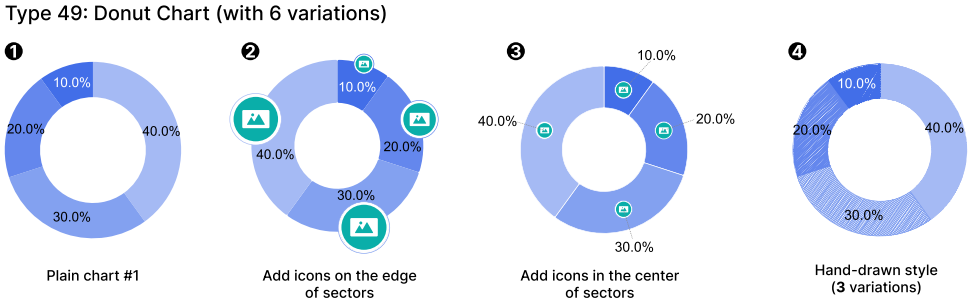


Figure 15: 75 chart types and 330 chart variations (Part 11).

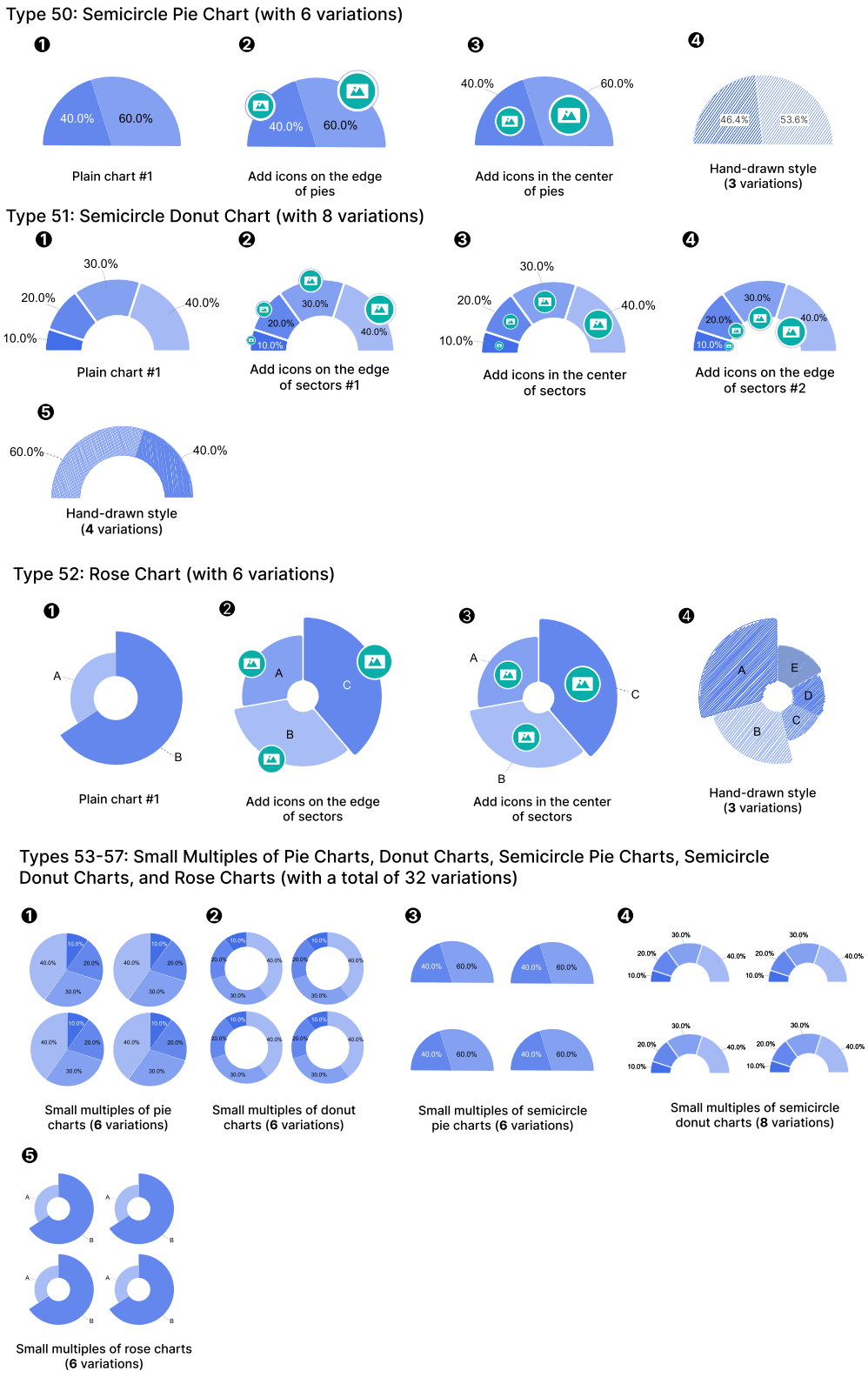


Figure 16: 75 chart types and 330 chart variations (Part 12).

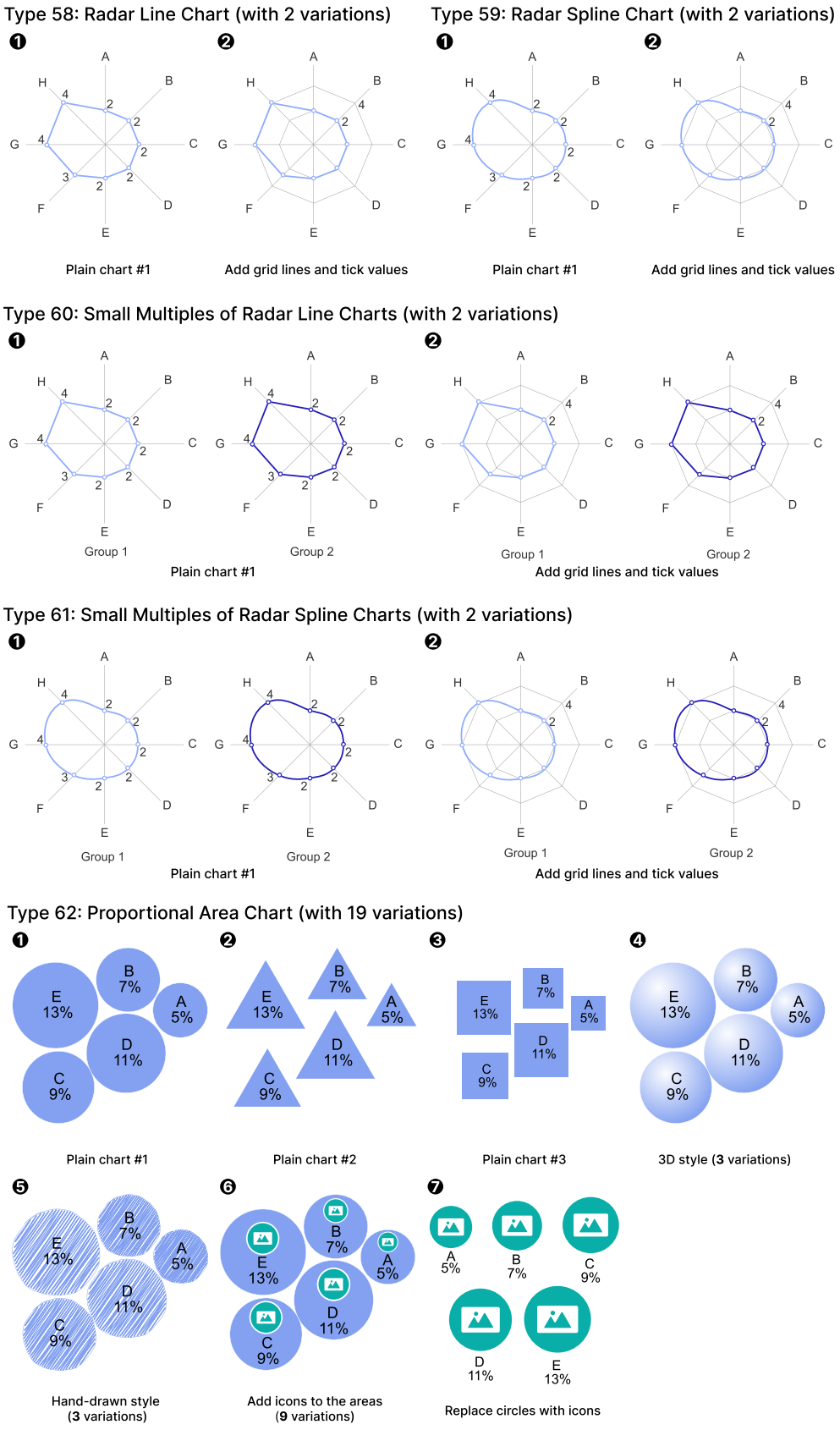


Figure 17: 75 chart types and 330 chart variations (Part 13).

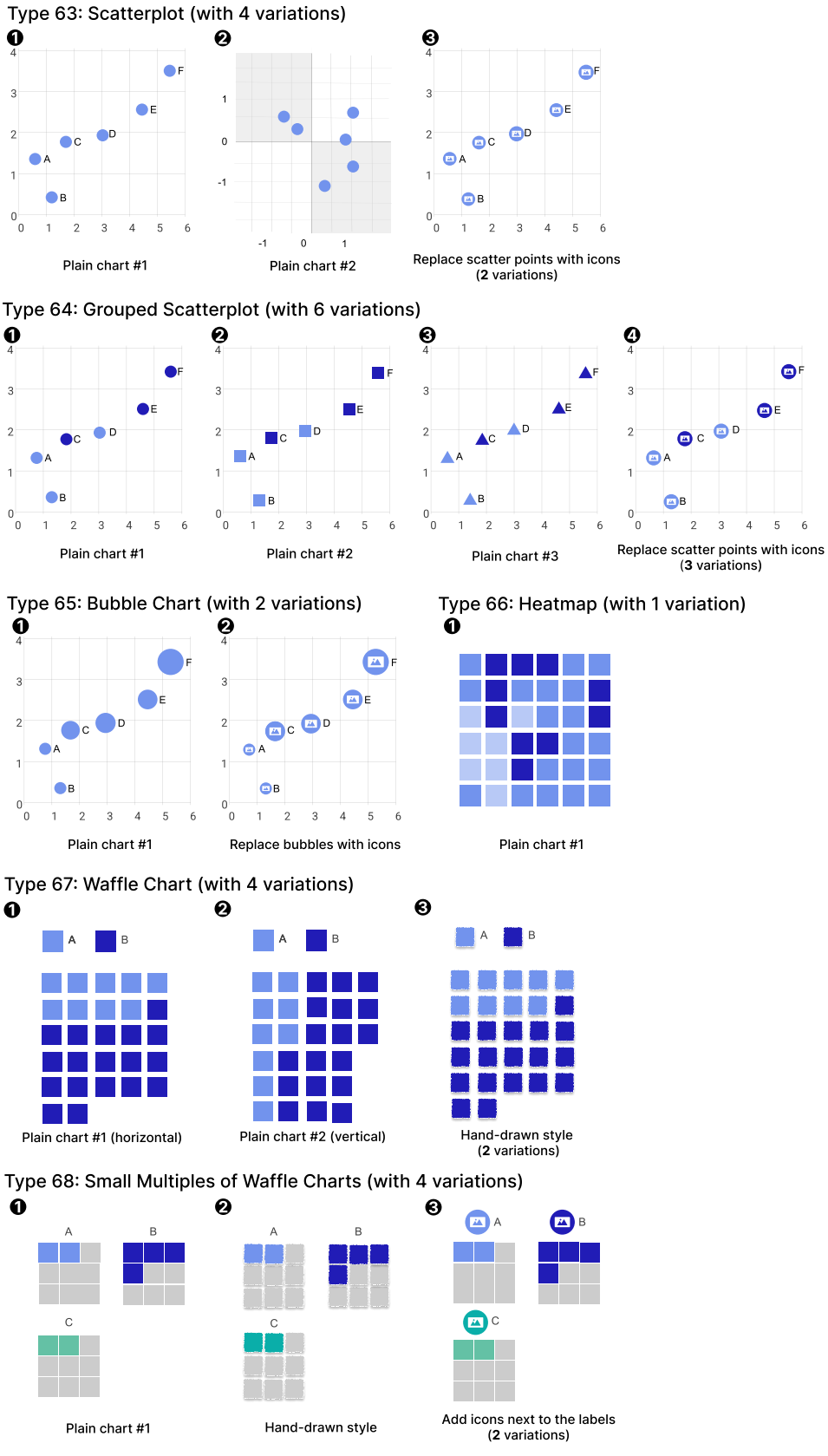
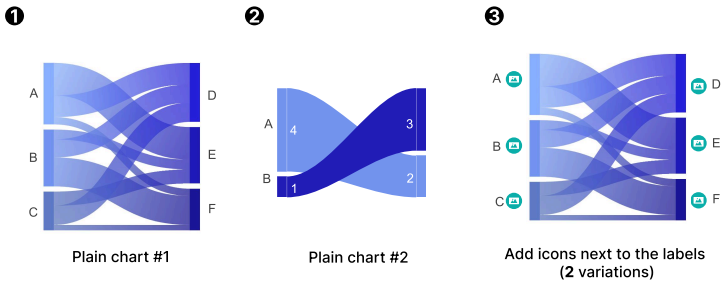
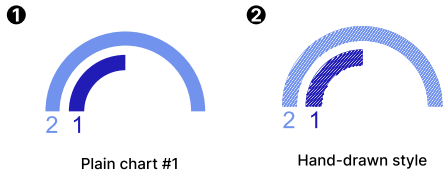


Figure 18: 75 chart types and 330 chart variations (Part 14).

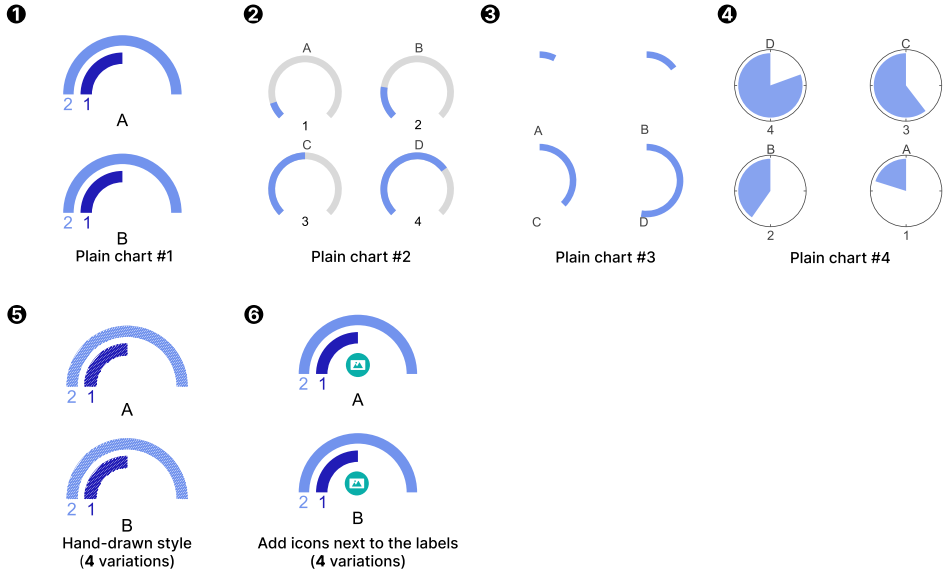
Type 69: Alluvial Diagram (with 4 variations)



Type 70: Gauge Chart (with 2 variations)



Type 71: Small Multiples of Gauge Charts (with 12 variations)



Type 72: Funnel Chart (with 4 variations)

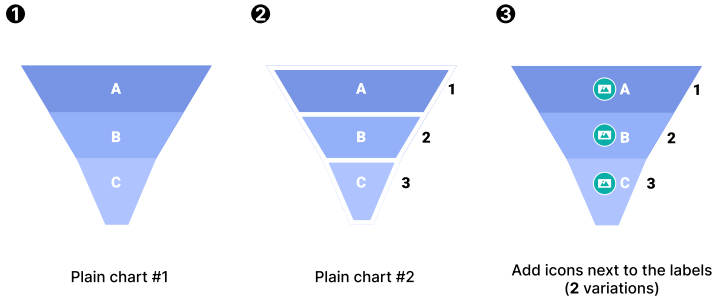
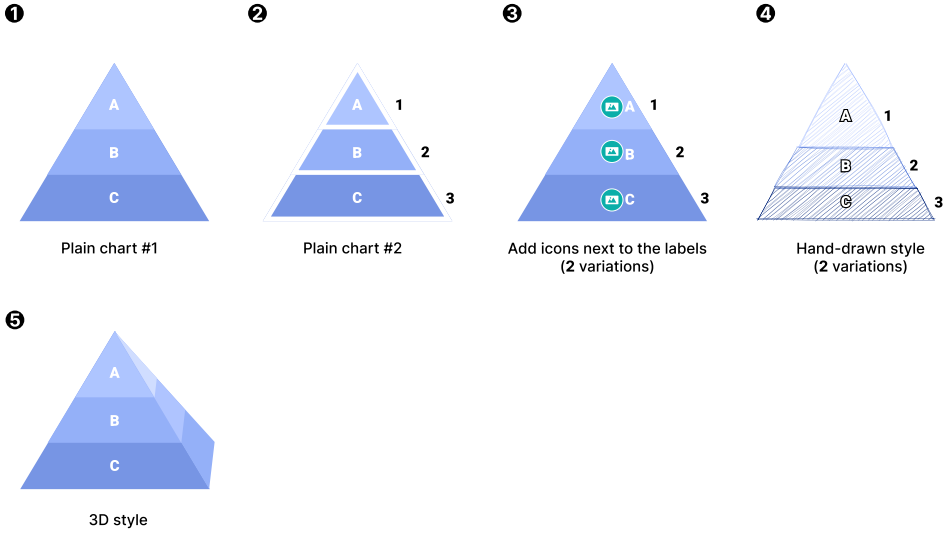
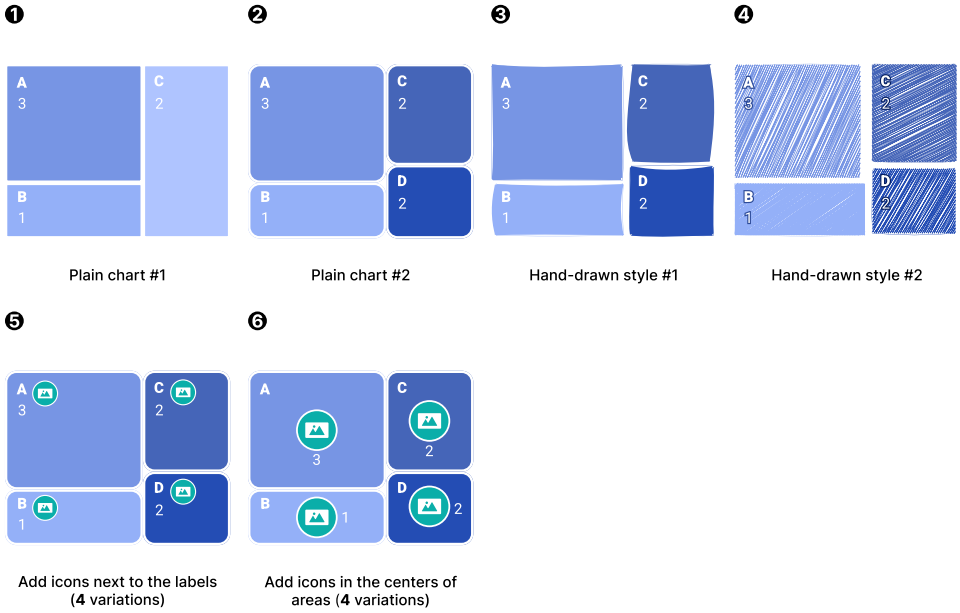


Figure 19: 75 chart types and 330 chart variations (Part 15).

Type 73: Pyramid Chart (with 7 variations)



Type 74: Treemap (with 12 variations)



Type 75: Voronoi Treemap (with 7 variations)

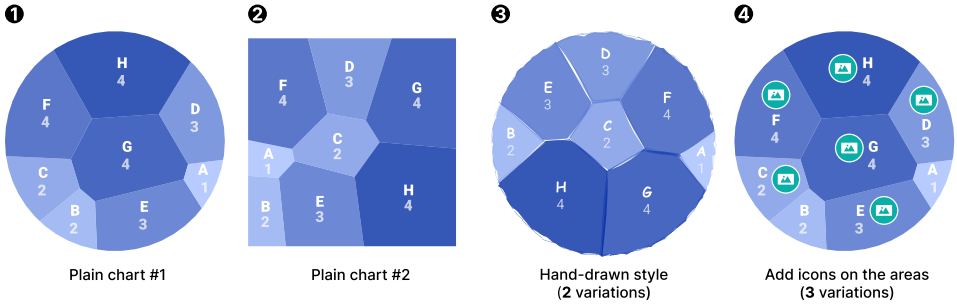


Figure 20: 75 chart types and 330 chart variations (Part 16).

C Layout Templates and Examples

C.1 Layout Templates

We summarize 68 layout templates from real infographic charts. The full list of these templates is in Figs. 21 and 22.

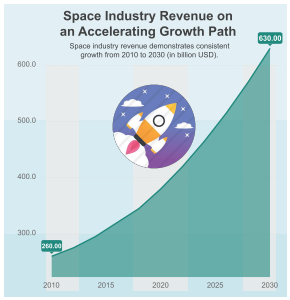
C.2 Examples

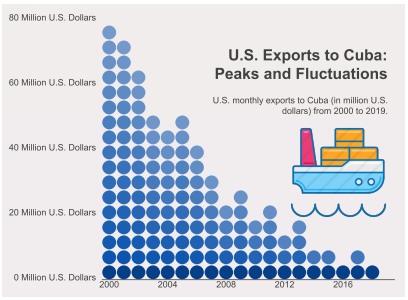
Figs. 23 and 24 consist of synthetic infographic chart examples that offer a quick preview of ChartGalaxy. To access the full dataset, please visit our dataset repository[1].

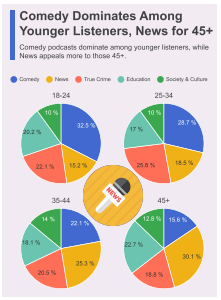
Figure 21: 68 layout templates (Part 1).

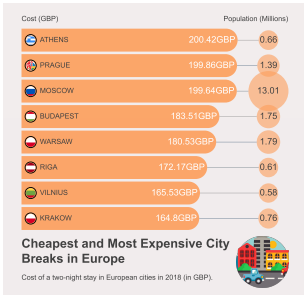
[1] https://huggingface.co/datasets/ChartGalaxy/ChartGalaxy

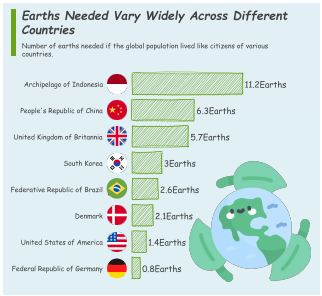
Figure 22: 68 layout templates (Part 2).

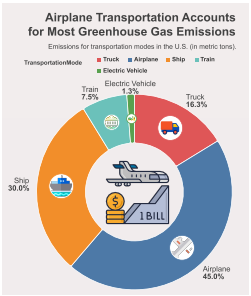


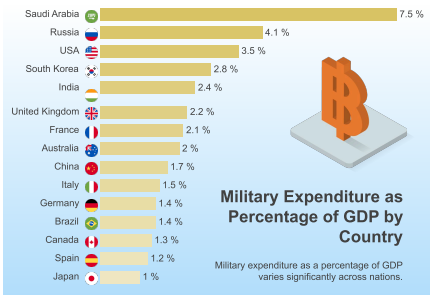


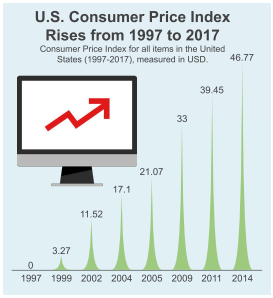


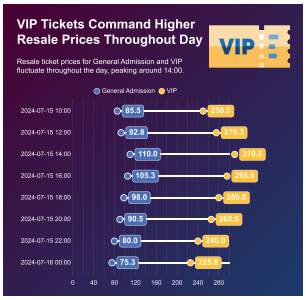












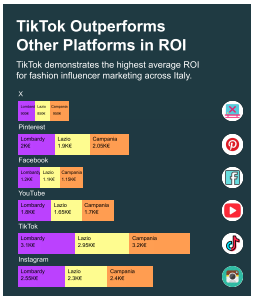


Figure 23: Synthetic infographic chart examples (Part 1).

Figure 24: Synthetic infographic chart examples (Part 2).

D Extended Evaluation and Results

D.1 Instruction Dataset for Infographic Chart Understanding

Table 7: All infographic chart understanding questions with definitions and examples included in our test set.

Question	Description	Example
Text-Based Reasoning		
Data Identification (DI)	Identify and report specific data values from the chart based on textual references.	Which Country has an AdoptionRate closest to 29.9? Select the correct answer from the following options: A. India, B. China, C. Germany, D. South Africa
Data Comparison (DC)	Compare multiple data points and their relationships to analyze relative values and derive insights.	What is the difference between the highest and lowest MedalCount? Please provide a numerical answer.
Data Extraction with Condition (DEC)	Extract specific data points from the chart that meet certain conditions.	What is the total PollutionLevel of Beijing in the given image? Please provide a numerical answer.
Fact Checking (FC)	Verify statements about data by cross-checking information against the visual representation.	Is the ApplicationRate of Spain consistently less than the ApplicationRate of Italy across all Years? Answer with exactly 'Yes' or 'No'.
Visual-Element-Based Reasoning		
Data Identification (DI)	Identify data values associated with specific visual elements (e.g., icons, symbols) in the chart.	What is the WaterConsumption for  in the Industrial group? Please provide a numerical answer.
Data Comparison (DC)	Compare data values associated with different visual elements, requiring cross-modal reasoning.	What is the difference between the AverageTeacherSalary of  and  ? Please provide a numerical answer.
Data Extraction with Condition (DEC)	Extract data values from visual elements when specific conditions are met.	What is the total Population of  in the given image? Please provide a numerical answer.
Fact Checking (FC)	Verify claims about data represented by visual elements, integrating visual and textual information.	Is the InterceptionRate of 2021 in  less than that in  ? Answer with exactly 'Yes' or 'No'.
Visual Understanding		
Style Detection (SD)	Identify design styles and formatting choices used in the chart.	What is the alignment style of the main text content (chart title and subtitle)? Select the correct answer from the following options: A. left-aligned, B. center-aligned, C. right-aligned, D. justified
Visual Encoding Analysis (VEA)	Analyze how data dimensions are encoded using visual properties (e.g., color, size, position, icons).	What data attribute or dimension is represented by icons in the given infographic chart? Select the correct answer from the following options: A. Location, B. WinPercentage, C. Team, D. None
Chart Classification (CC)	Identify the type of chart based on its visual characteristics and structure.	What types of charts are included in this infographic? Select the correct answer from the following options: A. Radial Grouped Bar Chart, B. Circular Grouped Bar Chart, C. Stacked Bar Chart, D. Diverging Bar Chart

Training Details We fine-tune two LVLMs in our experiments. The first is InternVL3-8B [65], which combines the Qwen2.5-7B language model with the InternViT-300M-448px-V2.5 visual

encoder. The second is Qwen2.5-VL-7B [66], which integrates the same Qwen2.5-7B language model with a Vision Transformer architecture optimized by Qwen. These models are fine-tuned on our instruction dataset for 2 epochs using a global batch size of 128. For parameter-efficient tuning, we utilize LoRA [81] with a rank of 8. This training process is conducted on 8 NVIDIA 4090D 48G GPUs, leveraging Deepspeed ZerO for parallelized training. For both InternVL3-8B and Qwen2.5-VL-7B, the learning rate is set to 5×10^{-5}, and we employ a cosine learning rate scheduler with a warm-up ratio of 0.1. The LoRA parameters, α and r, are both set to 32.

Evaluation Details Our test set is designed to evaluate chart understanding capabilities on infographic charts. It includes a variety of question types, which are categorized into text-based reasoning, visual-element-based reasoning, and visual understanding. Table 7 provides a detailed breakdown of these categories, including the definition and an illustrative example for each specific question type.

D.2 Benchmarking Infographic Chart Code Generation

Settings Table 8 lists the 17 LVLMs along with their API names used in this experiment. Since our task involves generating executable code with relatively long outputs, we conducted a small-scale pilot study to assess the basic code generation capability of each model. Based on this, we excluded models that consistently failed to produce meaningful or complete outputs under our task setting—for example, Phi-4—due to their limited capacity or inability to handle long sequences. We use greedy decoding (temperature $\tau = 0$) across all models to ensure deterministic outputs. To maximize the chance of obtaining complete and executable code, we configure each model to generate as many tokens as possible, setting the maximum generation length to $min(16384, A)$, where A denotes the model's maximum generation limit. This helps mitigate the risk of incomplete outputs, which result in non-executable code.

Table 8: API names of the evaluated LVLMs.

Model	Type	API name
Gemini-2.5-Pro [70]	Proprietary	`gemini-2.5-pro-preview-05-06`
Gemini-2.5-Flash [74]	Proprietary	`gemini-2.5-flash-preview-04-17`
Claude-3.7-Sonnet [72]	Proprietary	`claude-3-7-sonnet-20250219`
GPT-4.1 [71]	Proprietary	`gpt-4.1`
GPT-4.1-mini [71]	Proprietary	`gpt-4.1-mini`
GPT-4.1-nano [71]	Proprietary	`gpt-4.1-nano`
OpenAI-o4-mini [73]	Proprietary	`o4-mini`
OpenAI-o3 [73]	Proprietary	`o3`
OpenAI-o1 [75]	Proprietary	`o1`
GPT-4o [76]	Proprietary	`gpt-4o-2024-11-20`
Doubao-1.5-Vision-Pro [77]	Proprietary	`Doubao-1.5-vision-pro-32k`
Moonshot-v1-Vision [78]	Proprietary	`moonshot-v1-32k-vision-preview`
Llama-4-Maverick-17B [79]	Open-Source	`chutesai/Llama-4-Maverick-17B-128E-Instruct-FP8`
Llama-4-Scout-17B [79]	Open-Source	`chutesai/Llama-4-Scout-17B-16E-Instruct`
Qwen2.5-VL-72B [66]	Open-Source	`Qwen/Qwen2.5-VL-72B-Instruct`
Qwen2.5-VL-32B [66]	Open-Source	`Qwen/Qwen2.5-VL-32B-Instruct`
InternVL3-78B [65]	Open-Source	`internvl3-78b`

Benchmark details Our benchmark includes 75 chart types and 68 layout templates in ChartGalaxy. The associated tabular data contains an average of 15.02 data points. We also compute statistics on the number of SVG elements in all infographic charts, as this metric partially reflects the complexity of reproducing a given chart. On average, each chart contains 77.93 SVG elements, including 28.52 `text` elements and 8.07 `image` elements. Other commonly used visual elements include `rect` ($M = 24.36$), `circle` ($M = 6.21$), `path` ($M = 5.78$), and `line` ($M = 3.21$). Among all chart types, waffle charts are the most element-dense, with an average of 677.55 elements, while funnel charts are the simplest, with an average of only 14.60 elements.

Evaluation Metrics We present the details of the evaluation metrics of our benchmark, including the high-level score and the low-level score.

For the high-level score, we employ GPT-4o [76] to assess the visual similarity between the PNG image rendered by the generated code and the ground-truth one. We instruct GPT-4o to evaluate the similarity along six dimensions: data element, layout, text, image, color, and validity. The model

outputs a score for each of the six dimensions, which are then summed to produce a total score ranging from 0 to 100. The detailed prompt for this evaluation is provided in Supp. E.2.

The low-level score evaluates the fine-grained similarity between SVG elements of the rendered chart and the corresponding ground-truth chart. This evaluation is conducted through three steps: 1) decomposing both charts into SVG elements, 2) matching elements between the two charts, and 3) computing similarity metrics based on the matching results [67, 69]. Algorithm 1 presents the pseudo-code for the matching procedure. The full implementation details are available in our publicly accessible code repository[1].

Algorithm 1 SVG Element Matching Algorithm

Require: gt_leafs: Ground truth SVG elements.
Require: pr_leafs: Predicted SVG elements.
Require: $gt\_matched$: Array for ground truth matches (init with -1).
Require: $pr\_matched$: Array for prediction matches (init with -1).
Ensure: Updated matching information between elements.

$m \leftarrow |gt\_leafs|, n \leftarrow |pr\_leafs|$
$CostMatrix \leftarrow \text{ZeroMatrix}(m, n)$
for $i \leftarrow 0 \dots m-1$ **do**
  for $j \leftarrow 0 \dots n-1$ **do**
    $CostMatrix[i][j] \leftarrow \text{LeafCost}(gt\_leafs[i], pr\_leafs[j])$
  end for
end for
$(rows, cols) \leftarrow \text{HungarianAlgorithm}(CostMatrix)$ ▷ Returns optimal row-column pairs
for each pair (i, j) in $(rows, cols)$ **do**
  if $CostMatrix[i][j] \leq 1$ AND $gt\_matched[i] = -1$ AND $pr\_matched[j] = -1$ **then**
    $gt\_matched[i] \leftarrow j$
    $pr\_matched[j] \leftarrow i$
  end if
end for

Based on the matching results, the low-level score is computed as the average of six similarity metrics: area, text, image, color, position, and size. Let the parsed SVG elements of the ground-truth chart and the generated chart be denoted by $G = \{g_1, g_2, \dots, g_m\}$ and $P = \{p_1, p_2, \dots, p_n\}$, respectively, and let the set of matching pairs between G and P be M, where $(i, j) \in M$ indicates that g_i is matched with p_j. The detailed definitions and calculations of these metrics are provided below.

The area metric quantifies the proportion of the matched element areas relative to the total element areas:

$$\text{Area} = \frac{\sum_{(i,j)\in M} \big(S(g_i) + S(p_j)\big)}{\sum_{i=1}^{m} S(g_i) + \sum_{j=1}^{n} S(p_j)}, \tag{1}$$

where $S(\cdot)$ denotes the size of an element.

The text and image metrics evaluate the similarity of generated text and image elements, respectively, by averaging the similarity scores over all matched pairs of `text` and `image` elements. Unmatched `text` and `image` elements in ground-truth charts are assigned a similarity score of 0 to penalize generation failures. For `text` elements, similarity is computed using the character-level Sørensen-Dice coefficient, defined as twice the number of overlapping characters divided by the total number of characters in the two strings [67]. For `image` elements, similarity is measured using the CLIP embedding-based similarity [43].

The color, position, and size metrics assess visual consistency across matched elements with respect to their respective attributes. The color metric employs the CIEDE2000 formula [82] to measure the perceptual difference between the colors of matched elements. The position and size metrics are defined as follows:

$$\text{Position} = \frac{1}{|M|} \sum_{(i,j)\in M} \left[1 - \max\left(\left|X(g_i) - X(p_j)\right|, \left|Y(g_i) - Y(p_j)\right|\right)\right], \tag{2}$$

[1] `https://github.com/ChartGalaxy/ChartGalaxy`

where $(X(e), Y(e))$ denotes the normalized coordinates of the center of element e,

$$\text{Size} = \frac{1}{|M|} \sum_{(i,j)\in M} \left[1 - \frac{|S(g_i) - S(p_j)|}{\max\big(S(g_i), S(p_j)\big)} \right], \tag{3}$$

where $S(\cdot)$ denotes the size of an element.

Additional Analysis on LVLM performance We present additional results and analysis of LVLM performance on our benchmark, focusing on generated code length, performance across varying levels of complexity, and qualitative outcomes from the three top-performing models.

Generated code length and model performance. Table 9 reports the generated code length and overall performance of 17 LVLMs, with corresponding visualizations in Fig. 25. Code length is measured by token count using the GPT-2 tokenizer across all executable code segments. The statistics reveal significant variation in generated code length among different LVLMs. Among the 12 proprietary models, GPT-4.1-mini produces the longest code on average (7,656.71 tokens), surpassing both GPT-4.1 (5,237.78 tokens) and GPT-4.1-nano (2,387.05 tokens). This longer code length may partly explain GPT-4.1-mini's comparable performance to GPT-4.1. In contrast, Gemini-2.5-Pro and Gemini-2.5-Flash generate code of similar average length (6,662.45 vs. 6,508.22 tokens). OpenAI's o-series models produce the shortest average code length among top performers (2,886.75 for OpenAI-o4-mini, 2,662.11 for OpenAI-o1, and 3,114.74 for OpenAI-o3), which may reflect their ability to generate more concise solutions. Among the five open-source models, Qwen2.5-VL-32B has the longest average code length despite achieving the lowest overall score, while the remaining models exhibit comparable average lengths. These findings highlight the distinct coding styles of different LVLMs when generating extended code sequences.

Table 9: Generated code length and overall scores of different LVLMs. We measure code length in terms of tokens, utilizing the GPT-2 tokenizer.

Model	Length (AVG.)	Length (STD.)	Overall
Proprietary			
Gemini-2.5-Pro [70]	6,662.45	1,674.01	**85.21**
GPT-4.1 [71]	5,237.78	1,675.36	80.00
Claude-3.7-Sonnet [72]	5,870.50	1,552.48	79.91
GPT-4.1-mini [71]	**7,656.71**	**2,488.46**	79.69
OpenAI-o4-mini [73]	2,886.75	735.19	75.97
Gemini-2.5-Flash [74]	6,508.22	1,934.79	75.55
OpenAI-o1 [75]	2,662.11	1,018.11	74.69
OpenAI-o3 [73]	3,114.74	999.66	74.22
GPT-4o [76]	2,962.72	613.42	65.67
GPT-4.1-nano [71]	2,387.05	1,306.72	60.06
Doubao-1.5-Vision-Pro [77]	3,878.70	1,347.10	47.11
Moonshot-v1-Vision [78]	3,087.17	791.16	44.39
Open-Source			
Llama-4-Maverick-17B [79]	3,460.76	856.62	**61.29**
Qwen2.5-VL-72B [66]	3,512.20	1,236.29	57.09
InternVL3-78B [65]	3,376.52	758.71	55.07
Llama-4-Scout-17B [79]	3,247.25	826.56	51.91
Qwen2.5-VL-32B [66]	**4,960.82**	**1,264.43**	46.48

Model performance across different complexity levels. To comprehensively assess LVLM performance across varying degrees of difficulty, we divide our benchmark into three splits corresponding to different complexity levels. We adopt a straightforward heuristic based on the number of elements in an infographic chart to define complexity levels. While this simple approach does not consider layout complexity or chart types, it provides a practical and quantifiable basis for stratification, with more sophisticated measures left for future work. Based on this criterion, we split the benchmark into easy (240 infographic charts), medium (169), and hard (91) subsets. Fig. 26 presents the overall LVLM

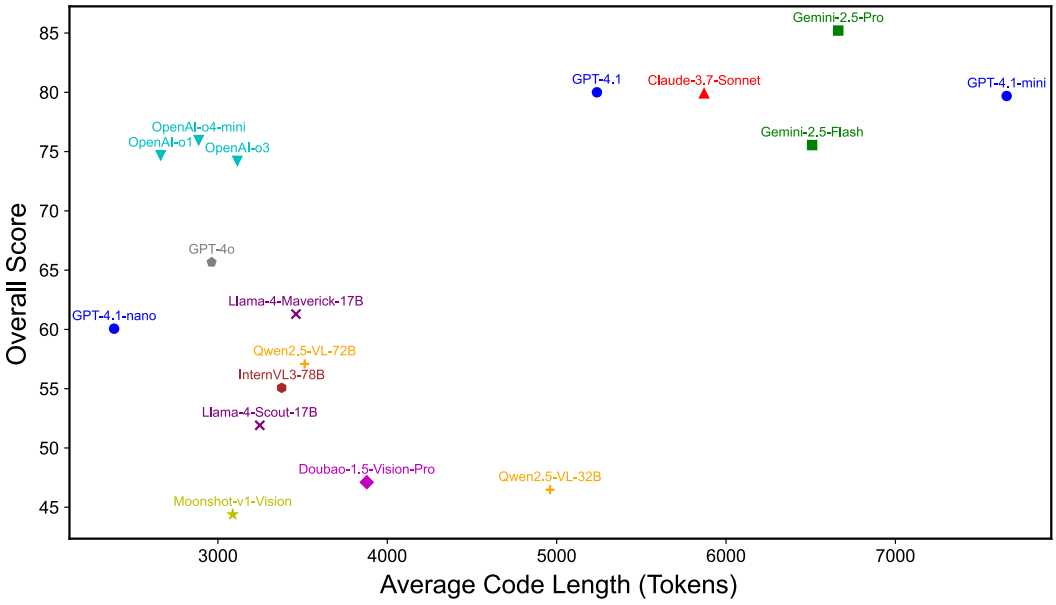


Figure 25: The average code length and overall scores of the evaluated LVLMs.

performance across these splits. Gemini-2.5-Pro consistently demonstrates superior performance at all levels of complexity. For most models, performance declines predictably as difficulty increases. Interestingly, Gemini-2.5-Flash and OpenAI-o3 exhibit improved results on the hard split, suggesting enhanced capability in understanding complex relationships among a large number of elements.

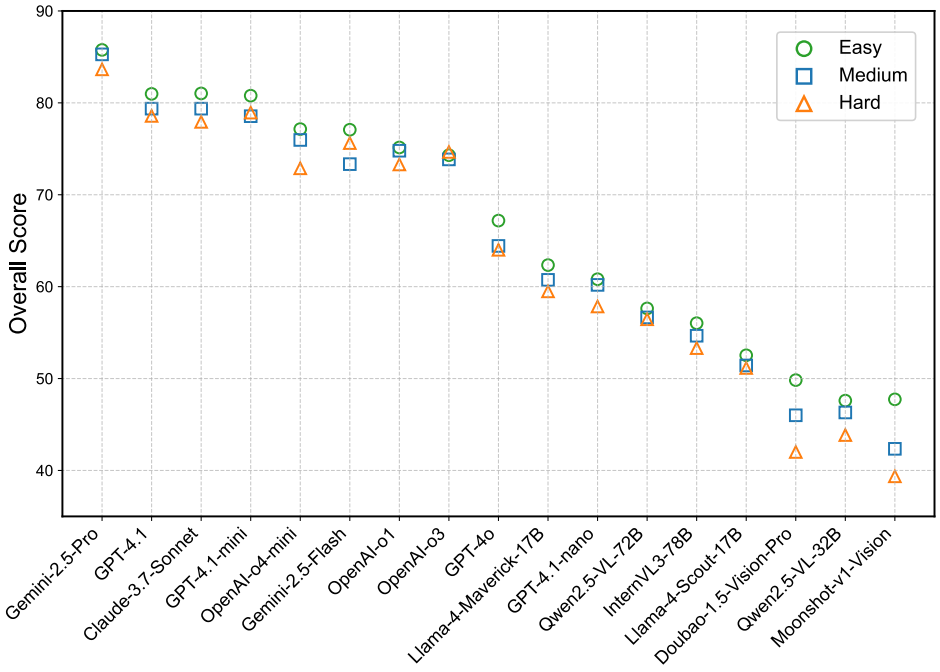


Figure 26: Overall scores of LVLMs across different complexity levels.

Qualitative examples. To illustrate model performance on our benchmark, we present five examples in Fig. 27, showing ground-truth charts alongside rendered ones from the generated code of the three top-performing LVLMs (Gemini-2.5-Pro, GPT-4.1, and Claude-3.7-Sonnet). These examples demonstrate

the models' relatively strong ability to generate code for infographic charts. Nevertheless, substantial room for improvement remains. Notably, for specialized chart variations (shown in the last four rows of Fig. 27), the models struggle to accurately reproduce spatial arrangement and data encoding.

Extended experiments We conduct two extended experiments to analyze the effects of different prompting methods and the thinking budget parameter on model performance. Both experiments are performed on a randomly sampled subset of 100 infographic charts from the benchmark dataset.

Prompting methods. In addition to the direct prompting method, we evaluate three alternative prompting strategies following prior work [39, 67]:

- **HintEnhanced**, which guides the model to focus on key aspects of the given infographic chart, such as layout, chart type, and data;
- **TableAug**, which provides auxiliary tabular data to the model;
- **SelfReflection**, where the model is provided the ground-truth chart, previously generated code from direct prompting, and the rendered chart, and is instructed to revise the given code.

Detailed prompts are available in our code repository[1]. The results, summarized in Table 10, show that the **SelfReflection** method consistently achieves the best performance across models.

Table 10: Performance comparison of LVLMs with different prompting methods.

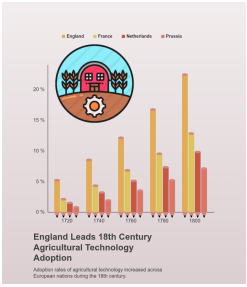
Model	Method	Exec. Rate	Low-Level							High-Level	Overall
			Area	Text	Image	Color	Position	Size	Avg.	GPT-4o	
Gemini-2.5-Pro [70]	Direct	**100.00**	**94.88**	**95.62**	86.95	88.46	**89.53**	68.09	87.25	83.59	85.42
	HintEnhanced	**100.00**	94.18	95.26	86.28	87.88	88.72	68.44	86.79	**84.36**	85.58
	TableAug	**100.00**	93.91	95.55	87.01	87.05	88.54	**68.75**	86.80	83.67	85.23
	SelfReflection	**100.00**	93.89	95.25	**88.68**	**88.51**	89.45	68.65	**87.40**	84.19	**85.80**
Llama-4-Maverick-17B [79]	Direct	**100.00**	78.24	58.51	59.53	66.35	74.00	47.74	64.06	58.42	61.24
	HintEnhanced	**100.00**	79.31	56.13	52.06	67.01	75.14	48.47	63.02	55.56	59.29
	TableAug	**100.00**	81.64	**61.24**	61.75	**68.66**	74.44	**48.97**	66.12	59.23	62.67
	SelfReflection	**100.00**	**82.44**	58.93	**66.19**	68.11	**75.45**	48.93	**66.67**	**59.46**	**63.06**

Thinking budget. Recent LVLMs have incorporated internal reasoning mechanisms that allow them to perform intermediate "thinking" steps prior to final output generation. This process is governed by the thinking budget parameter, which controls the token budget allocated for internal reasoning and is supported only by some reasoning-enabled models, such as Claude-3.7-Sonnet and Gemini-2.5-Flash. In our earlier experiments, we set the thinking budget to 1024 tokens for these models. For other reasoning models without explicit support for this parameter, we simulate the budget constraint by instructing them to limit internal thinking to 1024 tokens via prompt design. To investigate the effect of the thinking budget in greater detail, we conduct additional experiments varying this parameter for Claude-3.7-Sonnet and Gemini-2.5-Flash. As shown in Table 11, increasing the thinking budget for Claude-3.7-Sonnet leads to clear improvements in the image similarity metric and overall score. Conversely, for Gemini-2.5-Flash, a larger thinking budget correlates with declines in multiple metrics, including the area, text, and image metrics. These contrasting behaviors indicate that the impact of the thinking budget parameter on LVLM performance is model-dependent and warrants further investigation.

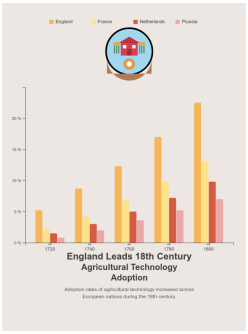
Table 11: Performance comparison of LVLMs with different thinking budgets.

Model	Thinking Budget	Exec. Rate	Low-Level							High-Level	Overall
			Area	Text	Image	Color	Position	Size	Avg.	GPT-4o	
Claude-3.7-Sonnet [72]	1024	**100.00**	92.29	**94.64**	78.82	85.55	87.28	66.62	84.20	76.89	80.55
	4096	**100.00**	92.91	93.28	82.42	86.42	88.10	**67.22**	85.06	76.12	80.59
	8192	**100.00**	**93.78**	93.85	**87.12**	**87.24**	**88.40**	**67.22**	**86.27**	**78.71**	**82.49**
Gemini-2.5-Flash [74]	1024	97.00	**90.18**	**89.60**	**78.51**	80.57	84.88	62.13	**80.98**	**75.69**	**78.34**
	4096	**98.00**	86.74	84.66	72.57	**82.33**	**85.91**	**64.59**	79.47	74.51	76.99
	8192	97.00	86.45	87.19	74.05	78.31	84.09	62.82	78.82	75.55	77.19

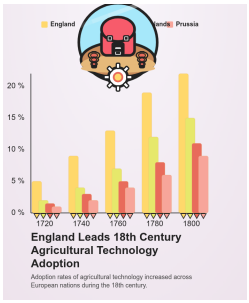
[1] `https://github.com/ChartGalaxy/ChartGalaxy`



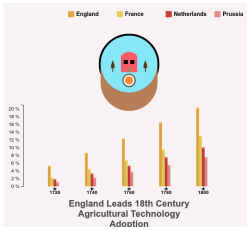
(a) Ground truth



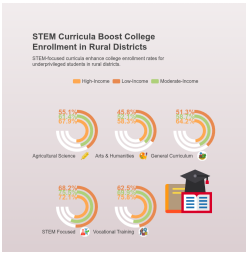
(b) Gemini-2.5-Pro



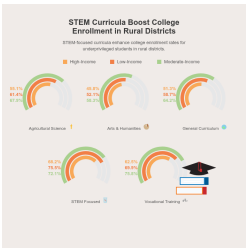
(c) GPT-4.1



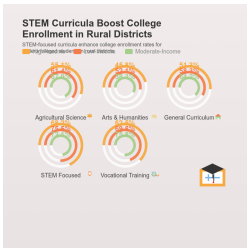
(d) Claude-3.7-Sonnet



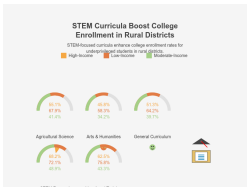
(e) Ground-truth



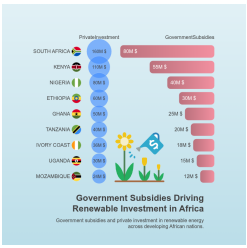
(f) Gemini-2.5-Pro



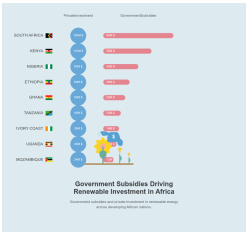
(g) GPT-4.1



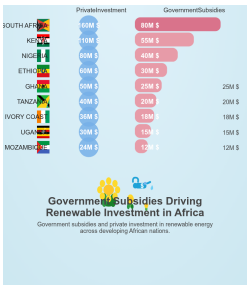
(h) Claude-3.7-Sonnet



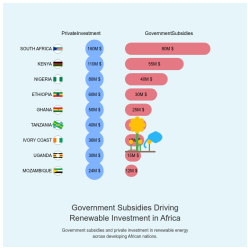
(i) Ground truth



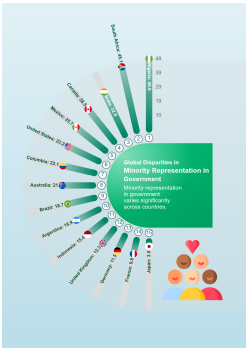
(j) Gemini-2.5-Pro



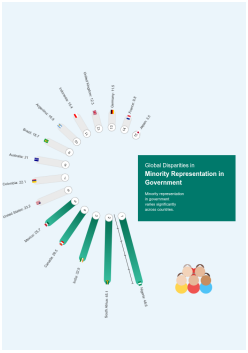
(k) GPT-4.1



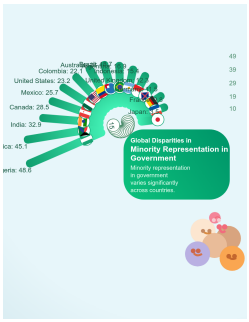
(l) Claude-3.7-Sonnet



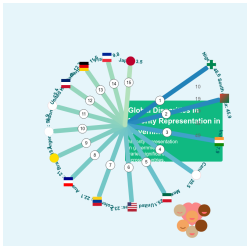
(m) Ground truth



(n) Gemini-2.5-Pro

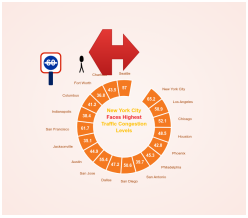


(o) GPT-4.1

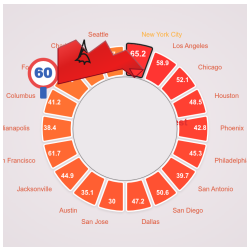


(p) Claude-3.7-Sonnet

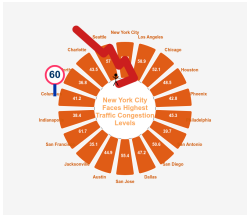
(q) Ground truth



(r) Gemini-2.5-Pro



(s) GPT-4.1



(t) Claude-3.7-Sonnet

Figure 27: Qualitative comparison of the generated infographic charts by Gemini-2.5-Pro, GPT-4.1, and Claude-3.7-Sonnet with the ground-truth ones.

D.3 Example-based Infographic Chart Generation

In the user study, each participant was compensated with 30 USD for their participation. Fig. 28 illustrates the user study interface using a specific example. Figs. 29-30 present all 30 triplets of infographic charts: one reference, two infographic charts to be rated that are generated by GPT-Image-1 and our method, respectively.

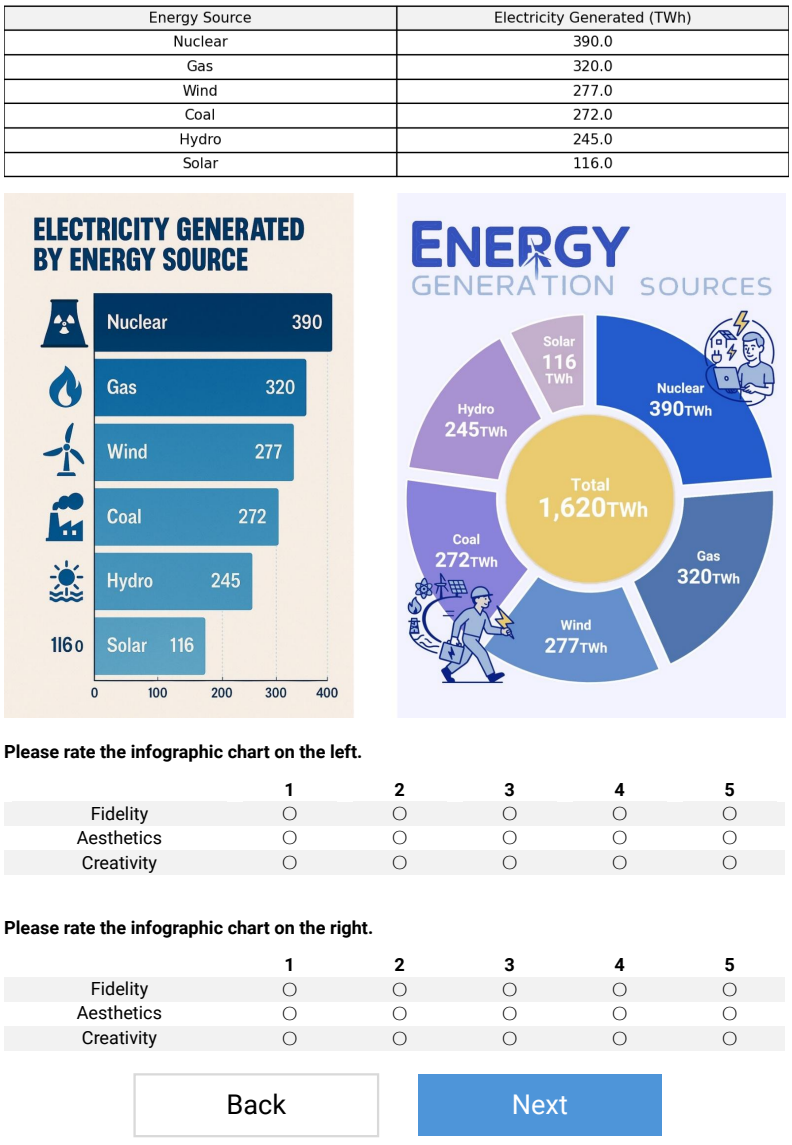


Figure 28: Screenshot of the user study interface. Participants were asked to compare two infographic charts generated by our method and GPT-Image-1 based on the same dataset and reference infographic chart, and rate each chart on three metrics: fidelity, aesthetics, and creativity, using a 5-point Likert scale.

Figure 29: Infographic charts used in the user study: 30 triplets, each comprising a reference, a GPT-generated, and a chart generated by our method (Part 1).

Figure 30: Infographic charts used in the user study: 30 triplets, each comprising a reference, a GPT-generated, and a chart generated by our method (Part 2).

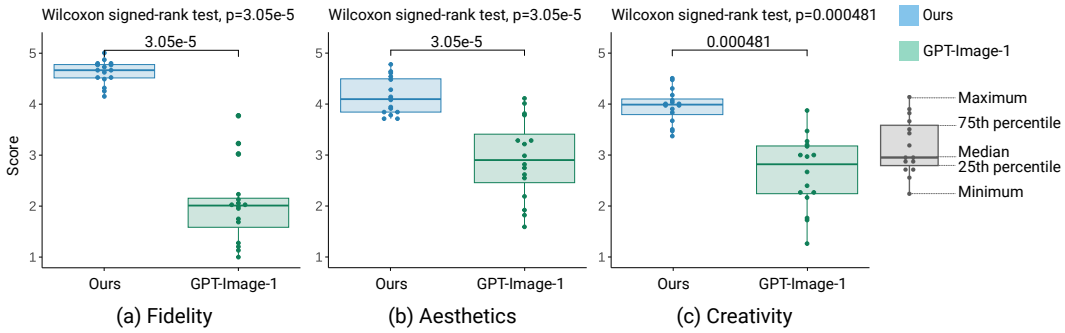


Figure 31: Performance comparison between our method and GPT-Image-1.

Fig. 31 presents the results of the Wilcoxon signed-rank test conducted on the user study data.

Fidelity. Ours significantly outperforms GPT-Image-1 in terms of fidelity. The average score for Ours is 4.63, compared to 2.10 for GPT-Image-1. The difference is statistically significant (V = 136, p = 3.05e-5), indicating that our method produces infographic charts that participants perceive as more accurate and faithful to the underlying data. This result can be attributed to our use of the template-based infographic chart creation method, which ensures numerical correctness and consistency between visual encodings and underlying values. In contrast, GPT-Image-1 often exhibits fidelity-related issues, such as incorrect labels, mismatched bar heights, and duplicated or omitted data elements.

Aesthetics. In terms of aesthetics, Ours receives an average score of 4.14, while GPT-Image-1 scored 2.90. The difference is statistically significant (V = 136, p = 3.05e-5). This difference is likely due to our method leveraging high-quality layout templates extracted from reference infographic charts, along with carefully designed color palettes to enhance visual harmony. In contrast, GPT-Image-1 occasionally produces unbalanced compositions or relies on overly simplistic and repetitive color palettes, reducing the overall visual appeal.

Creativity. Ours also achieves higher scores in creativity, with an average of 3.95 compared to 2.65 for GPT-Image-1. The difference is statistically significant (V = 136, p = 0.000481), suggesting that our method yields designs that are seen as more original and creative. This may be explained by our efforts to incorporate creative elements, such as embedding meaningful icons into the titles, and by exploring less conventional chart types beyond basic bar and line charts. In contrast, GPT-Image-1 tends to generate conventional titles and favors basic chart types, leading to lower perceived creativity.

E Prompts for Data Processing

E.1 Instruction Dataset for Infographic Chart Understanding

This section details the prompts utilized for constructing our instruction dataset for infographic chart understanding. The prompts generate question-answer pairs for both text-based reasoning and visual-element-based reasoning. These questions cover reasoning categories including Data Identification (DI), Data Comparison (DC), Data Extraction with Condition (DEC), and Fact Checking (FC). Visual-element-based reasoning questions are specifically created by determining if data attributes within a generated question possess corresponding icon representations in the chart image; if such icons exist, the textual attribute is then replaced by its respective icon. In addition to prompt-generated questions, we also incorporate template-based questions from ChartAssistant [38] as a supplement. For Visual Understanding questions, style-related questions (Style Detection, Visual Encoding Analysis, and Chart Classification) are derived from the chart styles used in our generation process and do not require specific prompts.

To generate question-answer pairs for Data Identification (DI), where the goal is to identify and report specific data values from the chart based on textual references, the following prompt is used:

 # DATA
{Tabular data}

Follow the data shown in the table strictly; keep answers concise and direct; avoid contradicting the table data.

 # INSTRUCTIONS
Generate straightforward Factoid Questions alongside their Corresponding Answers for the given image. The questions should focus on direct identification and extraction of explicit information such as specific data values, labels, titles, axis information, or quantities directly readable from the chart or its textual components. Avoid questions requiring inference, multi-step calculation, or comparison between multiple distinct data points. The Answers should be a number, text label, or a common phrase (Yes, No) found directly in the data. Respond in an Array of JSON objects format with the following keys: (i) Question, and (ii) Answer.

 # EXAMPLES
"According to the line graph, what was the 'Population' in 'New York' during '2010'?"
"What are the units indicated on the Y-axis of the 'Sales Performance' chart?"
"In the pie chart legend, which category does the color blue represent?"

For Data Comparison (DC) question-answer pairs, which involve comparing different data points or trends, this prompt is utilized:

 # DATA
{Tabular data}

Follow the data shown in the table strictly; keep answers concise and direct; avoid contradicting the table data.

 # INSTRUCTIONS
Generate some of the most difficult Factoid Questions alongside the Corresponding Answers for the given image. The questions could be related to numerical or visual reasoning. These questions should focus on making comparisons between different data points, categories, or time periods, and identifying significant differences or relationships between multiple elements in the data. The Answers could be a number, text label, or a common phrase (Yes, No). You should respond in an Array of JSON objects format with the following keys: (i) Question, and (ii) Answer.

 # EXAMPLES
"Which year had the highest gap between the headline inflation and core inflation?"
"In the years in which the red line was higher than the blue line, which year had the smallest difference between the red and green lines?"
"Which country had the highest increase in the number of cases between Jun and Jul?"
"Which country had the most significant drop in its share of the global hashrate between Aug 2021 and Sep 2021?"

For Data Extraction with Condition (DEC) question-answer pairs, which require extracting specific data points from the chart that meet certain conditions, the following prompt is used:

 # DATA
{Tabular data}

Follow the data shown in the table strictly; keep answers concise and direct; avoid contradicting the table data.

 # INSTRUCTIONS
Generate some of the most difficult Factoid Questions alongside the Corresponding Answers for the given image. The questions could be related to numerical or visual reasoning. These questions should focus on identifying trends, making comparisons, finding threshold crossings, analyzing patterns of change, or identifying significant events in the data. The Answers could be a number, text label, or a common phrase (Yes, No). You should respond in an Array of JSON objects format with the following keys: (i) Question, and (ii) Answer.

 # EXAMPLES
"Estimate the year in which wind capacity first exceeds 100 gw based on the trend shown in the chart."
"Determine the airline with the highest increase in ghg emissions from 2008 to 2014."
"How many times the retail sales growth went below the average annual percentage change from 2002 to 2010 by more than 2%?"
"Which event caused the most significant drop followed by quick recovery for both lines?"

The following prompt is used to facilitate Data Extraction with Condition (DEC) by generating questions that require calculations based on specific data points extracted from the chart under certain conditions:

 # DATA
{Tabular data}

Follow the data shown in the table strictly; keep answers concise and direct; avoid contradicting the table data.

 # INSTRUCTIONS
Generate some of the most difficult Factoid Questions alongside the Corresponding Answers for the given image. The questions could be related to numerical or visual reasoning. These questions should focus on performing calculations based on the data, such as computing percentages, averages, rates of change, or other mathematical operations on the values presented. The Answers could be a number, text label, or a common phrase (Yes, No). You should respond in an Array of JSON objects format with the following keys: (i) Question, and (ii) Answer.

 # EXAMPLES
"What is the average growth rate of renewable energy capacity between 2010 and 2015?"
"If the total investment in 2019 was $100 million, how much would be allocated to the healthcare sector based on the percentage shown?"
"Calculate the compound annual growth rate (CAGR) of smartphone sales from 2015 to 2020."

The following prompt is used for Data Extraction with Condition (DEC) in the context of hypothetical scenarios, where questions require extrapolations based on data points extracted under specific assumed conditions:

 # DATA
{Tabular data}

Follow the data shown in the table strictly; keep answers concise and direct; avoid contradicting the table data.

 # INSTRUCTIONS
You are an AI that generates concise and specific hypothetical questions based on chart images. Your task is to analyze the chart and generate a short, data-driven hypothetical question that explores future trends, impacts, or extrapolations based on the data. Avoid adding unnecessary explanations or context like "Based on the chart data..." or "A meaningful hypothetical question could be...". Keep the question focused and directly related to the chart. The question should make an assumption about future trends, impacts, or extrapolations based on the data.

 # EXAMPLES
"If the average wealth per person in Asia increases by 50%, what will be the new average wealth per person in Asia?"
"If the Construction index had stayed flat at its 2010 level throughout 2011-2013, would the overall Industry index likely have remained below its early 2011 peak?"
"If the Gini index continues to rise at the same rate as it did from 1980 to 2010, what will the Gini index be in 2025?"

The following prompt is designed for Fact Checking (FC), generating question-answer pairs that require verifying statements about data by cross-checking information against the visual representation in the chart:

 # DATA
{Tabular data}

Follow the data shown in the table strictly; keep answers concise and direct; avoid contradicting the table data.

 # INSTRUCTIONS
You are an AI that generates concise and specific factoid questions based on chart images. Analyze the given chart image and generate 2-3 pairs of claims and verdicts about its data. Half of the claims should be supported by the chart's data, while the other half are refuted. Avoid using terms like "rows", "columns", or "elements" from the data table; refer to "chart" or "chart image" instead. If the claim is supported, the verdict should be "True". If the claim is refuted, the verdict should be "False", followed by a brief explanation. The claims should cover comparisons of values or trends, basic statistical values (maximum, minimum, mean, median, mode) without using exact numbers from the chart. Ensure a diverse range of claims addressing various visual aspects of the chart, resulting in 2-3 turns of claims and verdicts. Generate the verdicts/answers without any additional explanation.

 # EXAMPLES
"Hong Kong consistently has the lowest percentages in at least three categories compared to other East Asian countries in the chart."
"The 4th grade reading pass rate at Auburn Elementary had improvement of about 8% from year 2014 to 2017."
"Toronto has the lowest average technology salary among the cities depicted in the chart."

E.2 Benchmarking Infographic Chart Code Generation

Prompt for instructing LVLMs We instruct the LVLMs to generate code for the provided infographic chart figure with the following prompt.

You are an expert data-visualization engineer and front-end developer.
Your task is to take a chart image and generate a HTML file that, when loaded in a browser, reproduces the chart exactly. The chart must be centered in the viewport.
 # Constraints:

- No Explanations: Do not include comments, reasoning, or explanatory text. Output only valid HTML with JavaScript code.

 # Technical Requirements

- **Charting library**: Use D3.js to implement the chart. Write the code to be clean, modular, and easy to understand and modify.
- **Single file output**: Provide one standalone HTML file that includes everything needed to render the chart.
- **Chart fidelity**: Replicate all visual elements—shapes, colors, axes, labels, legends, fonts, line weights, markers—exactly as in the original image.
- **No animations**: The chart must render immediately in its final state.
- **Aspect ratio & sizing**: The chart's content area (including margins, paddings, and plot area) must match the original image's proportions precisely.
- **Image**: Recreate any icon/image content using SVG `<g>` and shapes. Do not use `<image>`, base64 images, or links.
- **Text**: Place all text in `<text>` leaf nodes. Do not use `<tspan>` or nested text structures.

 # Grouping Requirements
You should use following `class` names for SVG groups with specific semantics:

- `title`: The title area (may contain title, subtitle, and shapes)
- `image`: Each individual icon/image/pictogram (no annotation text, no grouped images)
- `legend`: Legend area
- `axis`: Axis area

 # Output Format
Return only the following standalone HTML file:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <title>Recreated Chart</title>
        <!-- <style> -->
    </head>
    <body>
        <div id="chart-container"></div>
        <script src="https://d3js.org/d3.v7.min.js">
        </script>
        <script>
            <!-- Your D3 code here -->
        </script>
    </body>
</html>
```

Only output this file. Keep it minimal and strictly within the token limits.

Prompt for the high-level score We instruct GPT-4o to provide a high-level score with the following prompt.

You are an expert evaluator of visualizations. The first image (reference) is an infographic chart rendered from ground truth HTML code, while the second image is an infographic chart rendered from AI-generated HTML code. Your task is to assess how well the AI-generated chart replicates the reference chart.

 # Scoring Criteria (Total: 100 points):

1. **Data Element (20 points):**
   - Does the AI-generated chart accurately replicate all visual elements that encode data (e.g., bars, points, lines)?
   - The presence of extra or missing data elements that do not match the reference chart will negatively impact the score.
   - Are the positions and lengths/sizes of data elements consistent with the reference, such that the encoded values they represent appear similar?
2. **Layout (20 points):**
   - Does the layout of title, chart area, and images/icons in the AI-generated chart replicate the spatial arrangement of the reference chart?
   - Is alignment of the elements in the generated chart (e.g., left-aligned, right-aligned) consistent with that of the original chart?
   - Were element positions preserved, or did significant misalignments occur?
   - The white space inside the generated chart should be similar to the original, and the aspect ratio of the whole infographic chart should be preserved.
3. **Text (15 points):**
   - Does the AI-generated chart replicate all relevant text content accurately? This includes titles, axis labels, and annotations.
4. **Image (15 points):**
   - Does the AI-generated chart reproduce image elements from the reference infographic chart (e.g., thematic images, embedded icons)?
   - How visually similar are those image elements?
5. **Color (10 points):**
   - Does the AI-generated chart match the original one in terms of colors (background color, line colors, fill colors, text colors, etc.)? Minor differences due to rendering or anti-aliasing can be tolerated if the overall color scheme is preserved.
6. **Validity (20 points):**
   - Is the AI-generated chart clear, readable, and free of overlapping or occluded elements?
   - Are fundamental charting conventions followed? For example: Are axis ticks aligned with data, are icons placed near corresponding data, are colors in legends consistent with the data elements, and is axis-data correspondence preserved?

 # Evaluation Output Format (in JSON):
Please provide your evaluation in the following format (in valid JSON):

```
{
  "data_element": {
    "score": <integer>,
    "comment": "<your comment>"
  },
```

```
  "layout": {
    "score": <integer>,
    "comment": "<your comment>"
  },
  "text": {
    "score": <integer>,
    "comment": "<your comment>"
  },
  "image": {
    "score": <integer>,
    "comment": "<your comment>"
  },
  "color": {
    "score": <integer>,
    "comment": "<your comment>"
  },
  "validity": {
    "score": <integer>,
    "comment": "<your comment>"
  },
  "total_score": <sum of all scores above>
}
```

Be precise and detailed in your comments, and ensure the `total_score` equals the sum of all individual scores.

E.3 Example-based Infographic Chart Generation

You are an experienced infographic chart designer.
Given the following dataset in JSON format: {Tabular data} and a reference infographic chart image (used as a style guide) {Chart image}, your task is to create a new infographic chart that clearly and creatively visualizes the data.

 # INSTRUCTIONS
- Maintain overall stylistic consistency with the reference infographic chart, including color scheme, typography, iconography, and visual tone, to ensure a coherent aesthetic. However, adapt the layout, chart types, and visual elements creatively to best suit the structure and insights of the new dataset.
- Prioritize effective communication of the new data over replicating the original design.
- Incorporate visual storytelling elements, such as icons, labels, contrast, or scale, to highlight key patterns or contrasts in the data.
- Include a clear, well-designed title that matches the tone and aesthetic of the reference.
- You may choose a light or dark background — whatever best fits the visual narrative and legibility.
- Legends, axes, and any necessary annotations should be present and styled consistently.

 # OUTPUT FORMAT
- A single high-resolution infographic chart image (portrait or square format)
- All text and numbers should be **fully readable**
- The image should be **self-explanatory** — no external explanation should be needed
- The style should be **clean, professional, and ready for publication**

Avoid any explanation outside the visual — the final image should be self-explanatory and visually engaging.

F Ethical Considerations and Societal Impacts

Ethical consideration Our research involves the creation of a large-scale infographic chart dataset, comprising 1,151,087 synthesized charts and 104,519 real charts. We fully respect the intellectual property rights of original infographic designers. The real charts used in our dataset were all sourced from publicly available online platforms, and no proprietary or paywalled content was included. Where applicable, we extracted only the design patterns (chart variations and layout templates), not the original textual or commercial content, in adherence to fair use principles. Our proposed automatic generation method enables scalable generation of high-fidelity infographic-style charts, which can serve as a foundation for future research without infringing upon the rights of original creators.

Our project has been approved by our institution's internal ethics review. All human subjects involved in our user study have signed the user consent forms and have been provided with fair compensation in accordance with the local minimum wage standards.

Societal impacts Infographic charts are a powerful medium for communicating complex ideas to broad audiences. Our benchmark and generation pipeline support the development of LVLMs that can better understand and produce such visualizations, enhancing applications in education, journalism, and public communication. By enabling scalable, copyright-respecting data generation, our work lowers barriers for research while promoting ethical development. We believe this contributes meaningfully to improving the visual reasoning capabilities of MLLMs, especially in real-world tasks that require intuitive, data-driven storytelling.