# CHARTCRAFT: TOWARDS AESTHETIC AND CONTEXT-AWARE CHART GENERATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Structured data visualizations such as tables and charts are widely used to communicate information, yet existing chart generation datasets often suffer from limited aesthetics and insufficient contextual cues, hindering interpretability. We present **ChartCraft**, a framework and curated dataset designed to *advance aesthetic and context-aware* chart generation. Our approach focuses on transforming raw or minimally structured inputs into visually refined and context-enriched tables and charts that improve readability, semantic clarity, and visual coherence. To support this goal, we construct a diverse dataset spanning four visualization types (tables, bar, line, and pie charts), and define six generation tasks with paired prompts and image-level targets for visual styling and contextual augmentation. Building on this resource, we train a model termed **ChartCrafter** using SFT and RL, jointly modeling visual design and contextual reasoning for chart generation. We further introduce a new evaluation benchmark, **CraftEval**, to assess both aesthetic quality and contextual awareness. Experimental results demonstrate that models trained with ChartCraft produce charts that are more visually consistent and contextually informative, surpassing the baseline by +26.81 and +24.01 points in dimension-macro overall score, respectively on T2I and editing tasks of CraftEval. Our work highlights the importance of integrating design-aware and context-aware principles into aesthetic chart generation.

## 1 INTRODUCTION

Structured data visualizations, including tables and charts, play a central role in communicating quantitative information across scientific analysis, business intelligence, education, and public reporting (Masry et al., 2022; 2025a; Tang et al., 2025; Xia et al., 2025; Masry et al., 2023; Zhang et al., 2024a; Kahou et al., 2018). By organizing numerical data into such visual representations, complex patterns and relationships can be efficiently conveyed to human readers. As large data and models continue to grow rapidly, table and chart generation has become an increasingly important task, aiming to transform raw data, textual descriptions, or partially structured inputs into clear and interpretable visualizations. Recent advances in Large Language Models (LLMs) and generative models have further accelerated research in this field (Google, 2025b; Zhu et al., 2026; Wu et al., 2025a; Liu et al., 2025b; Esser et al., 2024; Rombach et al., 2022).

Chart-to-code methods (Wu et al., 2025b; Yang et al., 2025; Zhao et al., 2025; Zhang et al., 2024b; Hu et al., 2024; Masry et al., 2025b) provide executable, editable representations, while recent work also addresses image-based editing and infographic design (Yan et al., 2024; Li et al., 2025). We study complementary supervision for directly generating and editing chart images with aesthetic and contextual instructions. Pixel-space outputs accommodate illustrative styles, but require explicit data-fidelity checks and do not retain the exact editability of structured representations.

These two aspects are critical for effective visualization. Aesthetically coherent charts improve readability, visual hierarchy, and cognitive efficiency, allowing viewers to quickly identify salient patterns without unnecessary visual clutter. Meanwhile, contextual awareness, including informative titles, semantic labels, annotations, and design choices adapted to data characteristics, helps users correctly interpret the underlying message rather than merely observe graphical elements. Without aesthetic refinement and contextual grounding, automatically generated charts may remain technically



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




Figure 1: Comparison between ChartCraft and representative chart datasets in terms of scale, chart coverage, task diversity, and aesthetic/contextual supervision. ChartCraft provides 1.92M samples across four chart families and six generation/editing tasks, targeting design-aware and context-aware chart generation beyond code-centric synthesis.

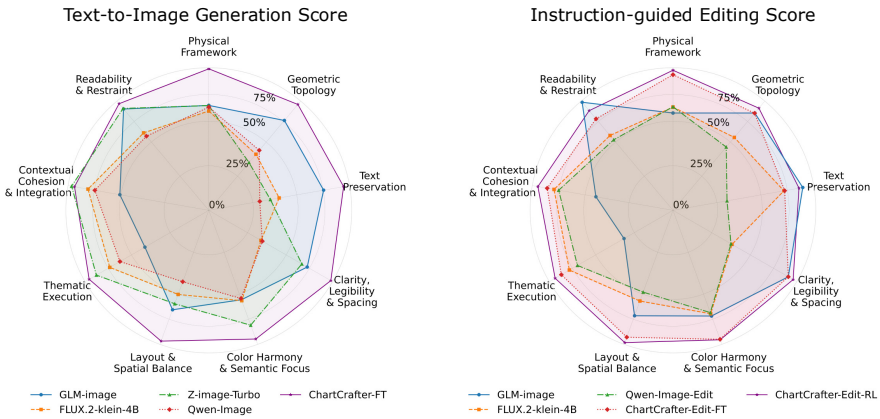


Figure 2: Fine-grained CraftEval scores of ChartCrafter and state-of-the-art open-source baselines across data integrity, aesthetic quality, and context/style alignment. ChartCrafter consistently improves over its Qwen-Image backbones, while RL further strengthens editing performance.

valid yet communicatively ineffective, limiting their practical usability in real-world analytical and reporting scenarios.

To address these limitations, we introduce **ChartCraft**, a curated dataset designed to advance aesthetic and context-aware chart generation, as illustrated in Figure 1.

The dataset spans four common visualization types—tables, bar charts, line charts, and pie charts—and covers six core generation and editing tasks: two text-to-image (T2I) generation tasks (Aesthetic Chart Generation and Contextual Chart Generation), and four instruction-guided editing tasks (Chart Aesthetic Refinement, Chart Contextualization, Aesthetic Chart Type Conversion, and Contextual Chart Type Conversion). The resulting image pairs and natural-language prompts enable models to jointly learn visual composition and semantic enhancement, treating chart generation as more than a structural problem.

Building upon this training resource, we develop ChartCrafter, a series of models trained using supervised fine-tuning (SFT) and reinforcement learning (RL) to integrate visual design reasoning with contextual decision-making during chart generation. To systematically measure progress beyond traditional correctness-based metrics, we further propose CraftEval, a new benchmark that evaluates both aesthetic quality and contextual alignment of generated visualizations.

CraftEval compares generated charts with reference images through three separate judge-based dimensions: Data & Structure Integrity (DSI), Aesthetic Quality Assessment (AQA), and Context & Style Alignment (CSA). These scores characterize generation quality; they do not certify exact numerical correctness.

Extensive experiments demonstrate that models trained with ChartCraft produce charts that are more visually consistent, semantically informative, and contextually appropriate compared to existing baselines (Figure 2).

Specifically, our fine-tuned model ChartCrafter-FT achieves a dimension-macro overall score of 70.49 on T2I tasks of CraftEval, surpassing the strong open-source baseline Qwen-Image (Wu et al., 2025a) (43.68) by +26.81 points. On the more challenging editing tasks, ChartCrafter-Edit-FT attains a dimension-macro overall score of 57.08, outperforming Qwen-Image-Edit-2509 (37.46) by +19.62 points, and ChartCrafter-Edit-RL further improves to a dimension-macro overall score of 61.47 (+24.01 over Qwen-Image-Edit-2509). Our findings suggest that incorporating design-aware and context-aware principles is important for building more communicative and practically useful automatic visualization systems.

In summary, this work makes the following contributions:

- We introduce **ChartCraft**, a curated dataset for aesthetic and contextual chart generation across multiple visualization types and generation tasks.
- We propose **ChartCrafter**, a training framework that jointly models visualization design and contextual reasoning using SFT and RL.
- We present **CraftEval**, a benchmark for evaluating aesthetic quality and contextual awareness in automatically generated charts.
- We demonstrate through extensive experiments that integrating design-aware and context-aware principles substantially improves the interpretability and usability of generated visualizations.

## 2 Related Work

### 2.1 Structured Generation and Chart Editing

Vega-Lite (Satyanarayan et al., 2017), Matplotlib (Hunter, 2007), and chart-to-code methods (Yang et al., 2025; Zhao et al., 2025) expose data and rendering parameters for precise modification. ChartReformer (Yan et al., 2024) already edits chart images without source data: it predicts tables and visual attributes and re-renders the result, covering style, layout, format conversion, and data changes. Its dataset is also named ChartCraft; it is distinct from ours. ChartEditor (Chen et al., 2026) predicts edited plotting code from an image and instruction, using SFT and RL with a rendering reward. Thus, image-input editing, type conversion, and SFT–RL are established capabilities. Our distinction is paired supervision for direct pixel-space generation and editing across six aesthetic/contextual tasks, adapting image models rather than predicting structured chart representations.

### 2.2 Infographic Generation and Evaluation

ChartGalaxy (Li et al., 2025) combines real and synthetic infographic charts with data tables, supporting understanding, code generation, and example-based infographic generation; aesthetic infographic design is therefore not unique to our setting. IGenBench (Tang et al., 2026) tests text-to-infographic reliability through atomic verification questions and reports both question-level and whole-image accuracy. CraftEval instead covers generation and editing, with separate DSI, AQA, and CSA scores; this task coverage does not imply more reliable correctness assessment. ChartBench (Xu et al., 2023) evaluates chart reasoning, including reading values without explicit data-point labels, whereas ChartQA (Masry et al., 2022) evaluates question answering. These understanding benchmarks probe numerical reading and are complementary to evaluating generated charts.

### 2.3 Scope of Our Contribution

We contribute a task-organized supervision resource and its evaluation protocol, with adaptation of existing image models (Wu et al., 2025a) as a use case. Our 1.92M training samples derive from 96,336 source images; this count is not directly comparable to unique charts or QA pairs. Nano Banana Pro supplies synthetic targets, so our results demonstrate task adaptation, not a

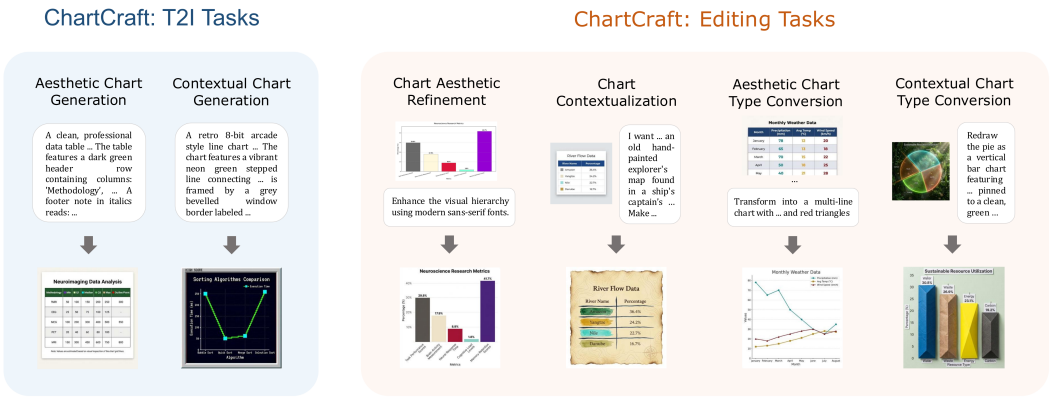


Figure 3: Data and Task Overview of ChartCraft Dataset, including two major subsets (T2I and Edit) targeting six tasks with distinct generation paradigms.

Table 1: ChartCraft dataset statistics. Prompt lengths are measured in words.

| Category | Property | Value |
|---|---|---|
| Scale | Source images (unique) | 96,336 |
| | Total training samples | 1,921,633 |
| | T2I samples | 531,648 |
| | Edit samples | 1,389,985 |
| Chart type | Table | 32% |
| | Bar | 32% |
| | Line | 23% |
| | Pie | 13% |
| Prompt length | Edit avg / median (words) | 22.9 / 22 |
| | T2I avg / median (words) | 103.5 / 103 |

new image-generation architecture or an established efficiency advantage over the teacher. Here, contextualization denotes instruction-conditioned visual presentation, including thematic styling; improved human comprehension requires separate evidence. Table 15 in Appendix H compares resource sizes, representations, and task scope.

# 3 METHOD

Our goal is to enable chart generation that is not only structurally correct but also aesthetically coherent and contextually informative. In this section, we describe three components: the construction of the ChartCraft dataset (Section 3.1), the training of the ChartCrafter model (Section 3.2), and the CraftEval benchmark for measuring visual and contextual quality (Section 3.3).

## 3.1 CHARTCRAFT DATASET

### 3.1.1 DATASET OVERVIEW

ChartCraft contains approximately 1.92 million high-quality chart–text paired samples organized around four visualization types: tables, bar charts, line charts, and pie charts. Detailed statistics are provided in Table 1. To overcome the visual monotony prevalent in existing chart datasets, ChartCraft adopts a ***diversity-first*** principle. Visual styles span from minimal professional designs to complex aesthetics drawn from a 23-category Visual Aesthetics Matrix, including cyberpunk, watercolor, retro-newspaper, glassmorphism, 3D claymation, and more. This diversity provides rich and varied supervision signals for models learning both aesthetic design and contextual reasoning.

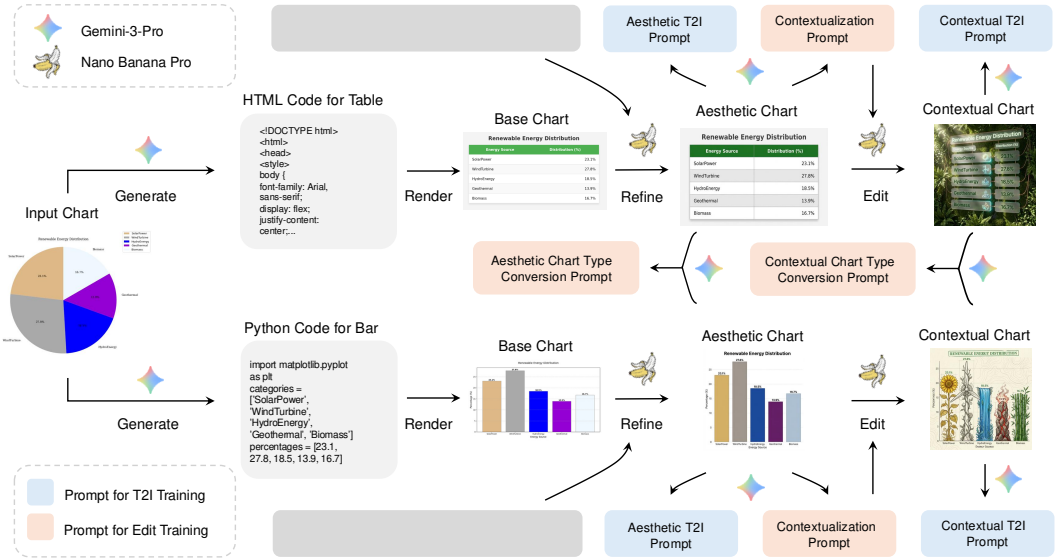


Figure 4: Curation pipeline of the ChartCraft dataset, containing six stages.

### 3.1.2 Task Definition

- The **T2I subset** (531K samples) focuses on direct chart generation from textual prompts and contains two tasks: *Aesthetic Chart Generation* (266K samples), which requires generating professionally styled charts with improved visual composition and layout, and *Contextual Chart Generation* (266K samples), which further incorporates semantic context such as informative titles, annotations, and context-aware design elements.
- The **Edit subset** (1,390K samples) focuses on instruction-guided chart editing and transformation, covering four tasks: *Chart Aesthetic Refinement* (284K samples), which improves the visual design of a given chart while preserving its underlying data and structure; *Chart Contextualization* (284K samples), which enriches an existing chart with contextual elements and styles; *Aesthetic Chart Type Conversion* (421K samples), which converts a chart into a different visualization type while maintaining consistent aesthetic styling and accurate data representation; and *Contextual Chart Type Conversion* (401K samples), which simultaneously transforms both the chart type and contextual presentation while preserving semantic consistency with the original data.

### 3.1.3 Curation Pipeline

To construct ChartCraft at scale with high quality, we design a six-stage automated curation pipeline, as illustrated in Figure 4.

- **Step 1: Feasibility Check & Code Generation.** We randomly select 100k charts from StructVisuals (Zhuo et al., 2025) as initial source candidates, which yield 96,336 unique source images after validity filtering and deduplication. Given a source chart image, Gemini-3-Pro (Google, 2025a) first assesses whether the image can be meaningfully converted into each of the four target chart types (table, bar, line, pie). For feasible conversions, the model extracts the underlying data and generates structured rendering code: HTML with embedded CSS for tables, and Python/Matplotlib (Hunter, 2007) scripts for the other chart types.
- **Step 2: Base Chart Rendering.** An automated code execution engine runs the generated scripts to produce raw chart renderings—structurally accurate but visually plain baseline images.
- **Step 3: Aesthetic Refinement.** The Nano Banana Pro (Google, 2025b) image editing API transforms each raw rendering into a visually polished chart by upgrading typography, applying professional color schemes, and removing visual clutter, while strictly preserving all data values and structural labels.

- **Step 4: Style Analysis & Instruction Synthesis.** To maximize style diversity across the dataset, we employ a ***Dual-Wheel*** synthesis mechanism: Gemini-3-Pro randomly pairs one of 5 user personas (e.g., executive, artist, analyst, geek, minimalist) with one of 23 visual aesthetics (e.g., cyberpunk, watercolor, retro-newspaper, 3D claymation) to generate tailored annotations for each aesthetic chart, including two components: a detailed *contextual instruction* guiding the generation model, and a concise *short prompt* simulating a natural user request.
- **Step 5: Contextual Augmentation.** Using the detailed contextual instruction from Step 4, Nano Banana Pro performs a second image editing pass on the aesthetic chart, injecting thematic materials, lighting effects, and contextual visual elements to produce the final context-enriched chart.
- **Step 6: Consistency Verification.** A Gemini-powered consistency checker compares the rendering code, aesthetic chart, and contextual chart to verify that all data values, labels, and chart types remain mutually consistent, filtering out any samples with hallucinations or structural corruption. This verification targets consistency among generated artifacts; we additionally audit the faithfulness of extracted rendering code to the original source images in Section A.6.

### 3.2 CHARTCRAFTER MODEL

Building upon the ChartCraft dataset, we train a chart generation model termed **ChartCrafter**, learning to generate charts that are aesthetically coherent while preserving structural correctness and semantic clarity. To achieve this objective, we adopt a two-stage training strategy consisting of supervised fine-tuning (SFT) and reinforcement learning (RL). SFT allows the model to learn fundamental chart generation and editing behaviors from the curated ChartCraft dataset, through which we obtain ChartCrafter-FT and ChartCrafter-Edit-FT, respectively for T2I and editing tasks. RL further optimizes the model toward higher aesthetic quality and contextual alignment, for which we specifically implement on the more challenging editing task, producing ChartCrafter-Edit-RL.

#### 3.2.1 SFT STAGE

The ChartCraft dataset provides large-scale, high-quality chart generation and editing data spanning diverse tasks and visual styles. SFT directly leverages this supervision: by training on curated examples, the model learns to faithfully follow aesthetic instructions, preserve data structure across style transformations, and produce context-enriched outputs that match the target distribution. Rather than training from scratch, we adapt two existing state-of-the-art diffusion models via Low-Rank Adaptation (LoRA) (Hu et al., 2022): `Qwen-Image` for text-to-image generation and `Qwen-Image-Edit-2509` for chart editing. This preserves the strong general visual priors of the base models while efficiently specializing them for chart-specific tasks. LoRA adapters (rank 32) are inserted into the Diffusion Transformer (DiT) backbone; all base model parameters are frozen throughout training. For the editing model we train on $\sim$50k image-editing pairs for 2 epochs; for the generation model we train on $\sim$50k text–image pairs for 5 epochs, using AdamW (Kingma & Ba, 2015) at $10^{-4}$ with bfloat16 mixed precision and gradient checkpointing.

#### 3.2.2 RL STAGE

While SFT provides a strong starting point, it is fundamentally an offline, imitation-learning objective: the model is trained on a fixed set of reference outputs and has no mechanism to explore beyond the observed data distribution. This especially matters for aesthetic and contextual generation of editing tasks, where the space of valid high-quality outputs is vast and a single ground-truth image captures only one point in it. *On-policy* reinforcement learning removes this constraint: the model generates its own outputs, receives reward feedback, and updates towards outputs that score higher, enabling it to discover and reinforce quality patterns that may not be well-represented in the static SFT data. Therefore, for our editing model, we apply Flow-GRPO (Liu et al., 2025a; Shao et al., 2024), a policy-gradient method adapted to continuous-flow diffusion models, starting from the SFT checkpoint, ChartCrafter-Edit-FT. We train only LoRA parameters and use $1024 \times 1024$ resolution throughout, with 10 diffusion steps during RL sampling and 50 steps for evaluation, a classifier-free guidance scale of 4, and bfloat16 mixed precision distributed via Accelerate with FSDP.

#### 3.2.3 REWARD DESIGN

A key design choice is the reward signal. Rather than using a proxy metric (e.g. FID (Heusel et al., 2017) or CLIP score (Hessel et al., 2021)) that may not correlate with actual chart quality, we use **ChartCraftScorer**, which directly wraps CraftEval's Gemini-based VQA evaluation. For each subtask, the applicable DSI, AQA, and CSA scores are selected, aggregated, and normalised to $[0, 1]$ as the RL return. This aligns the RL reward with the quality criteria used by CraftEval, while also introducing a possible judge-specific optimization risk. We therefore report independent human and external-judge validation in Section A.5.

### 3.3 CRAFTEVAL BENCHMARK

CraftEval targets aesthetic and contextual generation alongside data preservation. Executability, structural checks, and pixel similarity address different properties; executable code alone does not establish correct data encoding. We use separate judge-based dimensions to assess the generated images, with partial independent validation reported in Section A.5. To address these limitations, we introduce **CraftEval**, a dedicated benchmark designed to evaluate both the aesthetic quality and contextual awareness of generated charts. CraftEval provides carefully curated evaluation samples and a multi-dimensional scoring framework that jointly measures data fidelity, visual design, and style alignment.

#### 3.3.1 BENCHMARK OVERVIEW

CraftEval comprises 600 test samples equally distributed across 6 subtasks (100 samples each), covering both generation and editing scenarios. The **Edit** split contains 400 samples across four subtasks: Chart Contextualization (100), Aesthetic Chart Type Conversion (100), Chart Aesthetic Refinement (100), and Contextual Chart Type Conversion (100). The **T2I** split contains 200 samples across two subtasks: Aesthetic Chart Generation (100) and Contextual Chart Generation (100). Among the 6 subtasks, 3 involve style-conditioned evaluation — Chart Contextualization, Contextual Chart Type Conversion, and Contextual Chart Generation — and thus include an additional Context & Style Alignment (CSA) evaluation dimension alongside DSI and AQA. Each test sample is paired with a high-quality, human-vetted Ground Truth image serving as the reference standard for fine-grained comparative evaluation. We describe the split construction, source-image disjointness, and leakage checks in Section A.7.

#### 3.3.2 EVALUATION METRIC

CraftEval employs a Gemini-3-Pro MLLM-as-a-Judge protocol. For every test sample, the Ground Truth (Anchor) image is presented before the model-generated (Test) image to ensure consistent reference framing. Scoring follows a **dimension isolation** principle across three complementary dimensions, each scored via three sub-items on a 0/1/2 scale:

- **Data & Structure Integrity (DSI)** applies to all 6 subtasks and assesses whether the underlying data structure is preserved while ignoring visual style differences. Its sub-items are *Physical Framework*, *Geometric Topology*, and *Text Preservation*.
- **Aesthetic Quality Assessment (AQA)** applies to all 6 subtasks and measures visual refinement relative to the Ground Truth while ignoring data errors. Its sub-items are *Clarity, Legibility & Spacing*, *Color Harmony & Semantic Focus*, and *Layout & Spatial Balance*.
- **Context & Style Alignment (CSA)** applies only to the 3 style-conditioned subtasks (Chart Contextualization, Contextual Chart Type Conversion, and Contextual Chart Generation). Given a short style prompt, it evaluates style execution while ignoring structural errors. Its sub-items are *Thematic Execution*, *Contextual Cohesion & Integration*, and *Readability & Restraint*.

The overall score for each image is the mean of its applicable dimension scores (each in [0, 2]) multiplied by 50. Unless otherwise noted, overall gains in the text refer to the dimension-macro Overall scores in Tables 3 and 4; Table 2 reports per-subtask image-level averages.

Table 2: Per-subtask overall scores on CraftEval. T2I subtasks: *AestGen* (Aesthetic Chart Generation), *CtxGen* (Contextual Chart Generation). Edit subtasks: *AestRef* (Chart Aesthetic Refinement), *ChartCtx* (Chart Contextualization), *AestConv* (Aesthetic Chart Type Conversion), *CtxConv* (Contextual Chart Type Conversion). Per-subtask scores average only over applicable dimensions (CSA excluded for style-free subtasks).

| | T2I | | Edit | | | |
|---|---|---|---|---|---|---|
| **Model** | **AestGen** | **CtxGen** | **AestRef** | **ChartCtx** | **AestConv** | **CtxConv** |
| *Closed-source* | | | | | | |
| Nano Banana Pro | 89.36 | 84.78 | 78.36 | 79.00 | 62.61 | 66.66 |
| GPT Image 1.5 | 71.17 | 77.61 | 71.68 | 71.01 | 50.34 | 59.83 |
| Seedream 4.5 | 54.75 | 63.61 | 61.76 | 67.83 | 43.51 | 51.32 |
| *Open-source* | | | | | | |
| GLM-Image | 45.48 | 49.76 | 41.99 | 54.32 | 32.92 | 43.54 |
| FLUX.2-klein-4B | 23.96 | 43.08 | 43.74 | 53.58 | 16.20 | 34.70 |
| Z-Image-Turbo | 37.90 | 54.25 | – | – | – | – |
| Qwen-Image | 32.20 | 44.09 | – | – | – | – |
| ChartCrafter-FT | **66.68** | **67.37** | – | – | – | – |
| *Δ vs. Qwen-Image* | *+34.48* | *+23.28* | – | – | – | – |
| Qwen-Image-Edit-2509 | – | – | 39.49 | 43.37 | 16.45 | 32.86 |
| ChartCrafter-Edit-FT | – | – | **72.86** | 58.10 | 42.66 | 46.80 |
| *Δ vs. Qwen-Edit-2509* | – | – | *+33.37* | *+14.73* | *+26.21* | *+13.94* |
| ChartCrafter-Edit-RL | – | – | 71.71 | **67.68** | **44.51** | **49.54** |
| *Δ vs. Qwen-Edit-2509* | – | – | *+32.22* | *+24.31* | *+28.06* | *+16.68* |

# 4 Experiments

We evaluate ChartCrafter on CraftEval and summarize quantitative and qualitative results.

## 4.1 Experimental Settings

### 4.1.1 Implementation Details

All inference and evaluation experiments are conducted on NVIDIA H800 GPUs using HuggingFace Transformers (Wolf et al., 2019). Outputs are constrained to 1024×1024 pixels, except Seedream 4.5 (Seed, B., 2025), which is evaluated at 2048×2048 due to its minimum pixel requirement; a resolution-control analysis is reported in Section A.8. For all baselines, we follow each model's recommended inference parameters without manual tuning. T2I tasks receive only a text prompt, while Edit tasks receive a source image and editing instruction.

### 4.1.2 Compared Models

We compare ChartCrafter against state-of-the-art image generation and editing models. T2I baselines include Nano Banana Pro (Google, 2025b), GPT Image 1.5 (OpenAI, 2025), Seedream 4.5 (Seed, B., 2025), GLM-Image (Z.AI, 2025), FLUX.2-klein-4B (Labs, 2025), Z-Image-Turbo (Team, 2025), and Qwen-Image (Wu et al., 2025a). Edit baselines include Nano Banana Pro, GPT Image 1.5, Seedream 4.5, GLM-Image, FLUX.2-klein-4B, and Qwen-Image-Edit-2509. Results are summarized in Tables 2, 3, and 4.

## 4.2 Performance Analysis

**Dataset Effectiveness.** To isolate the contribution of the ChartCraft dataset, we compare ChartCrafter-FT against the Qwen-Image / Qwen-Image-Edit-2509 baseline using the same base architectures and inference settings. Since ChartCrafter uses LoRA adaptation on ChartCraft while the baselines are the original checkpoints, this comparison measures the end-to-end effect of ChartCraft supervision rather than a fully controlled data-only ablation. On T2I tasks, ChartCrafter-FT surpasses the Qwen-Image baseline by **+26.81** points in dimension-macro overall score. On the more challenging Edit tasks, the corresponding dimension-macro gain is **+19.62** points. Crucially, improvements are consistent across all three evaluation dimensions: on Edit tasks, DSI improves by +19.5, AQA by

Table 3: Fine-grained sub-item scores on CraftEval T2I tasks (scores in [0, 100]).

| Model | Overall | DSI | | | | AQA | | | | CSA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Overall | PF | GT | TP | Overall | CLS | CHS | LSB | Overall | TE | CCI | RR |
| *Closed-source* | | | | | | | | | | | | | |
| Nano Banana Pro | 87.9 | 85.5 | 90.2 | 89.0 | 77.2 | 84.0 | 82.0 | 89.2 | 80.8 | 94.0 | 89.0 | 98.0 | 95.0 |
| GPT Image 1.5 | 78.0 | 67.7 | 82.0 | 59.2 | 61.8 | 73.4 | 71.0 | 81.0 | 68.2 | 93.0 | 86.0 | 96.5 | 96.5 |
| Seedream 4.5 | 64.2 | 53.4 | 63.5 | 50.5 | 46.0 | 54.5 | 48.2 | 67.2 | 48.0 | 84.7 | 79.0 | 89.0 | 86.0 |
| *Open-source* | | | | | | | | | | | | | |
| GLM-Image | 51.5 | 39.8 | 57.2 | 33.2 | 28.7 | 46.2 | 41.5 | 54.2 | 42.8 | 68.5 | 60.0 | 61.0 | 84.5 |
| FLUX.2-klein-4B | 42.1 | 14.3 | 36.8 | 6.0 | 0.2 | 36.6 | 22.5 | 57.8 | 29.8 | 75.4 | 78.0 | 83.5 | 64.5 |
| Z-Image-Turbo | 53.8 | 27.2 | 48.5 | 19.2 | 14.0 | 50.0 | 43.5 | 64.8 | 41.8 | 84.2 | 79.0 | 87.0 | 86.5 |
| Qwen-Image | 43.7 | 29.4 | 50.0 | 24.5 | 13.8 | 36.3 | 27.8 | 53.5 | 27.8 | 65.4 | 60.0 | 71.0 | 65.0 |
| ChartCrafter-FT | **70.5** | **54.9** | **74.2** | **52.8** | **37.8** | **69.1** | **63.7** | **75.8** | **67.8** | **87.5** | **86.0** | **84.5** | **92.0** |
| *Δ vs. Qwen-Image* | *+26.8* | *+25.5* | *+24.2* | *+28.3* | *+24.0* | *+32.8* | *+35.9* | *+22.3* | *+40.0* | *+22.1* | *+26.0* | *+13.5* | *+27.0* |

Table 4: Fine-grained sub-item scores on CraftEval Edit tasks (scores in [0, 100]).

| Model | Overall | DSI | | | | AQA | | | | CSA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Overall | PF | GT | TP | Overall | CLS | CHS | LSB | Overall | TE | CCI | RR |
| *Closed-source* | | | | | | | | | | | | | |
| Nano Banana Pro | 74.2 | 74.3 | 80.1 | 78.1 | 64.5 | 62.8 | 63.8 | 58.2 | 66.2 | 85.4 | 78.0 | 88.5 | 89.8 |
| GPT Image 1.5 | 67.8 | 53.7 | 72.8 | 44.6 | 43.9 | 61.3 | 62.9 | 57.1 | 63.8 | 88.3 | 79.8 | 90.2 | 94.8 |
| Seedream 4.5 | 59.9 | 54.1 | 65.6 | 51.4 | 45.4 | 50.1 | 52.5 | 45.8 | 52.1 | 75.5 | 62.5 | 76.8 | 87.2 |
| *Open-source* | | | | | | | | | | | | | |
| GLM-Image | 45.3 | 40.5 | 43.2 | 47.0 | 31.4 | 45.4 | 50.0 | 39.5 | 46.5 | 49.9 | 23.8 | 37.2 | 88.8 |
| FLUX.2-klein-4B | 42.0 | 34.3 | 47.4 | 30.6 | 25.1 | 32.1 | 21.0 | 38.2 | 37.0 | 59.5 | 58.8 | 67.5 | 52.2 |
| Qwen-Image-Edit-2509 | 37.5 | 27.9 | 47.6 | 25.5 | 10.8 | 30.0 | 20.6 | 37.6 | 31.8 | 54.5 | 52.2 | 63.8 | 47.5 |
| ChartCrafter-Edit-FT | 57.1 | 47.5 | 70.1 | 46.9 | 25.4 | 54.2 | 50.0 | 52.2 | 60.4 | 69.6 | 65.2 | 73.0 | 70.5 |
| *Δ vs. Qwen-Edit-2509* | *+19.6* | *+19.5* | *+22.5* | *+21.4* | *+14.6* | *+24.2* | *+29.4* | *+14.6* | *+28.6* | *+15.1* | *+13.0* | *+9.2* | *+23.0* |
| ChartCrafter-Edit-RL | **61.5** | **51.2** | **73.2** | **50.2** | **30.1** | **56.4** | **52.9** | **52.5** | **63.9** | **76.8** | **70.2** | **80.5** | **79.5** |
| *Δ vs. Qwen-Edit-2509* | *+24.0* | *+23.3* | *+25.6* | *+24.7* | *+19.3* | *+26.5* | *+32.3* | *+14.9* | *+32.1* | *+22.3* | *+18.0* | *+16.7* | *+32.0* |

+24.2, and CSA by +15.1. The fact that all three complementary dimensions benefit simultaneously indicates that ChartCraft provides effective supervision signals for structural preservation, aesthetic refinement, and style adherence alike. Among the three, the AQA gain is the largest, reflecting how the dataset's diverse visual styles and rich aesthetic annotations translate directly into stronger aesthetic reasoning in the trained model.

**Effect of Reinforcement Learning.** Building on the SFT checkpoint, Flow-GRPO further improves ChartCrafter-Edit-FT to ChartCrafter-Edit-RL, yielding an additional **+4.39** points in dimension-macro overall score. The per-dimension breakdown reveals a clear pattern: CSA benefits most from RL (+7.2), followed by DSI (+3.7) and AQA (+2.2). This is consistent with the on-policy nature of the training: SFT largely saturates the more objective structural criteria (DSI) that are well-covered by the training distribution, while the subjective aesthetic and style dimensions (CSA, AQA) retain room for improvement through reward-guided self-exploration. At the subtask level, RL is especially effective on Chart Contextualization (+9.57), the task that requires generating context-enriched, style-adherent outputs — precisely the scenario where CSA reward feedback is most informative. These results confirm that SFT and RL play complementary roles: SFT establishes broad task coverage, while RL refines the model's output toward the evaluation criteria that are hardest to capture through imitation alone.

**Additional Analyses.** Appendix A reports additional validation, leakage checks, extended chart-type pilots, limitations, and failure modes; qualitative comparisons are in Appendix B.

## 5 Conclusion

We introduced **ChartCraft**, a large-scale dataset for aesthetic and context-aware chart generation, together with **ChartCrafter** models trained by SFT and RL and **CraftEval** for multi-dimensional evaluation. Results on CraftEval show consistent gains over strong open-source baselines, highlighting the value of design-aware and context-aware supervision for automatic visualization.

## AI USE STATEMENT

Generative AI was used for synthetic dataset construction, code and instruction generation, chart editing, consistency filtering, and automated evaluation and reward computation, as described in Section 3 and the appendix. An LLM assistant also supported literature retrieval, drafting and revising the related-work discussion, feedback on evaluation methodology, and LaTeX formatting. The validation procedures and limitations of the research pipeline are described in the corresponding sections.

## REFERENCES

Liangyu Chen, Yichen Xu, Jianzhe Ma, Yuqi Liu, Donglu Yang, Liang Zhang, Zihao Yue, Wenxuan Wang, and Qin Jin. ChartEditor: A reinforcement learning framework for robust chart editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 2026. doi: 10.1609/aaai.v40i24.39107. URL https://arxiv.org/abs/2511.15266.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR / OpenReview.net, 2024.

Google. Gemini 3 pro model card. https://deepmind.google/models/model-cards/gemini-3-pro/, 2025a. Accessed: 2026.

Google. Introducing nano banana pro. https://blog.google/innovation-and-ai/products/nano-banana-pro/, 2025b. Accessed: 2026.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP (1)*, pp. 7514–7528. Association for Computational Linguistics, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, pp. 6626–6637, 2017.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net, 2022.

Linmei Hu, Duokang Wang, Yiming Pan, Jifan Yu, Yingxia Shao, Chong Feng, and Liqiang Nie. Novachart: A large-scale dataset towards chart understanding and generation of multimodal large language models. In *ACM Multimedia*, pp. 3917–3925. ACM, 2024.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3): 90–95, 2007. doi: 10.1109/MCSE.2007.55.

Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Ákos Kádár, Adam Trischler, and Yoshua Bengio. Figureqa: An annotated figure dataset for visual reasoning. In *ICLR (Workshop)*. OpenReview.net, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Zhen Li, Duan Li, Yukai Guo, Xinyuan Guo, Bowen Li, Lanxi Xiao, Shenyu Qiao, Jiashu Chen, Zijian Wu, Hui Zhang, Xinhuan Shu, and Shixia Liu. ChartGalaxy: A dataset for infographic chart understanding and generation. *arXiv preprint arXiv:2505.18668*, 2025. URL https://arxiv.org/abs/2505.18668.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online RL. *CoRR*, abs/2505.05470, 2025a.

Zhihang Liu, Xiaoyi Bao, Pandeng Li, Junjie Zhou, Zhaohe Liao, Yefei He, Kaixun Jiang, Chen-Wei Xie, Yun Zheng, and Hongtao Xie. Showtable: Unlocking creative table visualization with collaborative reflection and refinement. *CoRR*, abs/2512.13303, 2025b.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq R. Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *ACL (Findings)*, volume ACL 2022 of *Findings of ACL*, pp. 2263–2279. Association for Computational Linguistics, 2022.

Ahmed Masry, Parsa Kavehzadeh, Do Xuan Long, Enamul Hoque, and Shafiq Joty. Unichart: A universal vision-language pretrained model for chart comprehension and reasoning. In *EMNLP*, pp. 14662–14684. Association for Computational Linguistics, 2023.

Ahmed Masry, Mohammed Saidul Islam, Mahir Ahmed, Aayush Bajaj, Firoz Kabir, Aaryaman Kartha, Md. Tahmid Rahman Laskar, Mizanur Rahman, Shadikur Rahman, Mehrad Shahmohammadi, Megh Thakkar, Md. Rizwan Parvez, Enamul Hoque, and Shafiq Joty. Chartqapro: A more diverse and challenging benchmark for chart question answering. In *ACL (Findings)*, volume ACL 2025 of *Findings of ACL*, pp. 19123–19151. Association for Computational Linguistics, 2025a.

Ahmed Masry, Abhay Puri, Masoud Hashemi, Juan A. Rodríguez, Megh Thakkar, Khyati Mahajan, Vikas Yadav, Sathwik Tejaswi Madhusudhan, Alexandre Piché, Dzmitry Bahdanau, Christopher Pal, David Vázquez, Enamul Hoque, Perouz Taslakian, Sai Rajeswar, and Spandana Gella. Bigcharts-r1: Enhanced chart reasoning with visual reinforcement finetuning. *CoRR*, abs/2508.09804, 2025b.

OpenAI. Gpt-image-1.5. `https://openai.com/zh-Hans-CN/index/new-chatgpt-images-is-here/`, 2025. Accessed: 2026.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10674–10685. IEEE, 2022.

Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. Vega-lite: A grammar of interactive graphics. *IEEE Trans. Vis. Comput. Graph.*, 23(1):341–350, 2017.

Seed, B. Seedream 4.5. `https://seed.bytedance.com/en/seedream4_5`, 2025. Accessed: 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024.

Liyan Tang, Grace Kim, Xinyu Zhao, Thom Lake, Wenxuan Ding, Fangcong Yin, Prasann Singhal, Manya Wadhwa, Zeyu Leo Liu, Zayne Sprague, Ramya Namuduri, Bodun Hu, Juan Diego Rodriguez, Puyuan Peng, and Greg Durrett. Chartmuseum: Testing visual reasoning capabilities of large vision-language models. *CoRR*, abs/2505.13444, 2025.

Yinghao Tang, Xueding Liu, Boyuan Zhang, Tingfeng Lan, Yupeng Xie, Jiale Lao, Yiyao Wang, Haoxuan Li, Tingting Gao, Bo Pan, Luoxuan Weng, Xiuqi Huang, Minfeng Zhu, Yingchaojie Feng, Yuyu Luo, and Wei Chen. IGenBench: Benchmarking the reliability of text-to-infographic generation. *arXiv preprint arXiv:2601.04498*, 2026. URL `https://arxiv.org/abs/2601.04498`.

Z-Image Team. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Shengming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *CoRR*, abs/2508.02324, 2025a.

Chengyue Wu, Zhixuan Liang, Yixiao Ge, Qiushan Guo, Zeyu Lu, Jiahao Wang, Ying Shan, and Ping Luo. Plot2code: A comprehensive benchmark for evaluating multi-modal large language models in code generation from scientific plots. In *NAACL (Findings)*, volume NAACL 2025 of *Findings of ACL*, pp. 3006–3028. Association for Computational Linguistics, 2025b.

Renqiu Xia, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Botian Shi, Junchi Yan, and Bo Zhang. Chartx and chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *IEEE Trans. Image Process.*, 34:7436–7447, 2025.

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

Pengyu Yan, Mahesh Bhosale, Jay Lal, Bikhyat Adhikari, and David Doermann. ChartReformer: Natural language-driven chart image editing. In *International Conference on Document Analysis and Recognition*, pp. 453–469. Springer, 2024. doi: 10.1007/978-3-031-70533-5_26. URL https://arxiv.org/abs/2403.00209.

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, Gongye Liu, Xiaomei Nie, Deng Cai, and Yujiu Yang. Chartmimic: Evaluating lmm's cross-modal reasoning capability via chart-to-code generation. In *ICLR*. OpenReview.net, 2025.

Z.AI. GLM-Image. https://github.com/zai-org/GLM-Image, 2025. Accessed: 2026.

Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. Tinychart: Efficient chart understanding with program-of-thoughts learning and visual token merging. In *EMNLP*, pp. 1882–1898. Association for Computational Linguistics, 2024a.

Songheng Zhang, Lei Wang, Toby Jia-Jun Li, Qiaomu Shen, Yixin Cao, and Yong Wang. Chartifytext: Automated chart generation from data-involved texts via LLM. *CoRR*, abs/2410.14331, 2024b.

Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Zhiyuan Liu, and Maosong Sun. Chartcoder: Advancing multimodal large language model for chart-to-code generation. In *ACL (1)*, pp. 7333–7348. Association for Computational Linguistics, 2025.

Dawei Zhu, Rui Meng, Yale Song, Xiyu Wei, Sujian Li, Tomas Pfister, and Jinsung Yoon. Paperbanana: Automating academic illustration for AI scientists. *CoRR*, abs/2601.23265, 2026.

Le Zhuo, Songhao Han, Yuandong Pu, Boxiang Qiu, Sayak Paul, Yue Liao, Yihao Liu, Jie Shao, Xi Chen, Si Liu, and Hongsheng Li. Factuality matters: When image generation and editing meet structured visuals. *CoRR*, abs/2510.05091, 2025.

# APPENDIX

## A ADDITIONAL VALIDATION, LIMITATIONS, AND RELEASE PLAN

### A.1 SUBTASK DIFFICULTY ANALYSIS

Averaging scores across all models reveals a consistent difficulty ordering among the four Edit subtasks: Aesthetic Chart Type Conversion is the hardest (mean score 38.65), followed by Contextual Chart Type Conversion (48.15), Chart Aesthetic Refinement (60.20), and Chart Contextualization (61.86). Aesthetic Chart Type Conversion demands simultaneous satisfaction of structural fidelity (the data must be re-encoded into a different chart type) and aesthetic quality (the visual style must be preserved), placing the highest joint requirement on DSI and AQA. By contrast, Chart Aesthetic Refinement and Chart Contextualization benefit from explicit style prompts that guide the model's generation, reducing the search space and making them comparatively more tractable.

### A.2 DIMENSION COMPLEMENTARITY

Table 5: Pairwise correlation between CraftEval dimension scores. CSA is only applicable to the 300 samples that include a style prompt.

| **Dimension Pair** | $n$ | **Pearson** $r$ | **Spearman** $\rho$ | **Kendall** $\tau$ |
|---|---|---|---|---|
| DSI $\leftrightarrow$ AQA | 600 | 0.484 | 0.488 | 0.404 |
| DSI $\leftrightarrow$ CSA | 300 | 0.181 | 0.188 | 0.152 |
| AQA $\leftrightarrow$ CSA | 300 | 0.432 | 0.445 | 0.363 |

### A.3 HUMAN–GEMINI AGREEMENT

Table 6: Pairwise ordering consistency between human expert scores and Gemini-based CraftEval scores on 80 annotated samples.

| **Granularity** | **Ordered pairs** | **Consistency** |
|---|---|---|
| Sample total | 6,320 | 76.4% |
| All sub-items | 385,020 | 74.2% |
| DSI sub-items | 57,360 | 80.6% |
| AQA sub-items | 57,360 | 69.9% |
| CSA sub-items | 19,740 | 68.6% |

### A.4 HUMAN EVALUATION PROTOCOL

For the human–Gemini agreement study and the blind preference study, evaluators were shown anonymized chart images without model names. For agreement scoring, evaluators rated the same DSI, AQA, and CSA sub-items used by CraftEval on a three-level scale corresponding to fail, partial, and pass. For blind preference, evaluators compared paired outputs against the same input or instruction and selected the chart with better data preservation, readability, and contextual alignment, with ties allowed. No personal or demographic information was collected from evaluators.

### A.5 INDEPENDENT HUMAN AND EXTERNAL-JUDGE VALIDATION

Because Gemini-3-Pro is used in both CraftEval and the RL reward, we evaluate the editing models with an additional protocol that is independent of the Gemini-based scorer. Table 7 reports blind human preference and Qwen3.5-27B-based judging on the same held-out samples. This analysis checks whether the ranking observed under CraftEval is preserved under evaluators that are not used for reward optimization.

Table 7: Independent validation on held-out CraftEval samples. Human preference is measured by blind pairwise comparison against the baseline, with percentages averaged over all human judgments. Qwen3.5-27B is not used in data filtering, reward design, or CraftEval scoring.

| Model | Samples | Human Win | Human Tie | Qwen3.5-27B Overall | Rank vs. CraftEval |
|---|---|---|---|---|---|
| Qwen-Image-Edit-2509 | 100 | – | – | 36.8 | 3 |
| ChartCrafter-Edit-FT | 100 | 68.5% | 12.0% | 55.4 | 2 |
| ChartCrafter-Edit-RL | 100 | 78.0% | 9.5% | 62.1 | 1 |

### A.6 Source-Code Faithfulness Audit

Step 6 verifies that generated artifacts remain mutually consistent with the extracted rendering code. To assess the fidelity of the intermediate representation, we audit whether the extracted code preserves the original source image. Table 8 reports chart-type, label, and numerical-value agreement on a manually checked sample.

Table 8: Audit of Gemini-extracted rendering code against the original source charts.

| Audit Set | Samples | Chart Type Acc. | Label Match | Value Match |
|---|---|---|---|---|
| Table | 100 | 100.0% | 98.6% | 99.2% |
| Bar | 100 | 100.0% | 97.4% | 95.8% |
| Line | 100 | 99.0% | 95.5% | 93.4% |
| Pie | 100 | 100.0% | 98.1% | 96.7% |
| All | 400 | 99.8% | 97.4% | 96.3% |

### A.7 Split Independence and Leakage Checks

To assess benchmark independence, we verify that CraftEval has no overlap with the ChartCraft training split at four levels: source image IDs, exact text prompts, perceptual image duplicates, and chart data signatures. Table 9 summarizes these split-integrity checks.

Table 9: Split-integrity checks between ChartCraft training data and CraftEval. No overlap is found under source-ID, prompt, perceptual-image, or chart-data-signature matching.

| Check | Train | CraftEval | Overlap / Duplicates |
|---|---|---|---|
| Source image IDs | 96,336 | 600 | 0 |
| Exact text prompts | 1,921,633 | 600 | 0 |
| Perceptual image duplicates | 1,921,633 | 600 | 0 |
| Chart data signatures | 1,921,633 | 600 | 0 |

### A.8 Resolution-Control Analysis

Seedream 4.5 is evaluated at $2048 \times 2048$ because its API requires that minimum resolution. To quantify the effect of this resolution difference, Table 10 reports a control experiment in which Seedream outputs are downsampled to the common $1024 \times 1024$ evaluation resolution before scoring.

Table 10: Resolution-control analysis for Seedream 4.5. Scores are computed with the same CraftEval protocol.

| Seedream Setting | T2I Overall | Edit Overall | T2I Delta | Edit Delta |
|---|---|---|---|---|
| Native $2048 \times 2048$ | 64.2 | 59.9 | – | – |
| Downsampled $1024 \times 1024$ | 62.6 | 58.4 | -1.6 | -1.5 |

### A.9 Pilot on Additional Chart Types

ChartCraft focuses on four common visualization types that cover a large fraction of everyday analytical communication. To assess extensibility beyond this scope, Table 11 reports a pilot evaluation on more complex charts. These results provide an applicability check rather than a full-scale benchmark for these additional chart types.

Table 11: Pilot applicability test on additional chart types. Feasibility is reported as an averaged evaluator score over the pilot set.

| Chart Type | Samples | Feasibility Score | DSI | AQA |
|---|---|---|---|---|
| Scatter plot | 50 | 78.5% | 66.4 | 70.2 |
| Stacked bar chart | 50 | 88.0% | 76.8 | 74.5 |
| Multi-panel / subplot | 50 | 42.6% | 51.3 | 58.7 |
| Heatmap | 50 | 65.4% | 62.1 | 68.9 |

### A.10 Failure Modes

We summarize failure modes in Table 12. The main categories we track are numerical/text corruption, over-stylization that harms readability, and chart-type conversion errors. Counts are reported over 200 T2I samples and 400 Edit samples from CraftEval. This analysis characterizes remaining limitations of ChartCrafter.

Table 12: Failure-mode summary on CraftEval.

| Failure Mode | T2I Count (n=200) | Edit Count (n=400) |
|---|---|---|
| Numerical or label corruption | 15 | 31 |
| Unreadable / overlapping text | 9 | 22 |
| Over-stylized context | 6 | 18 |
| Chart-type conversion error | – | 27 |

### A.11 Limitations and Release Plan

ChartCraft is designed as a dataset and benchmark for aesthetic and context-aware generation over four common chart families. It does not yet cover the full space of visualization designs, such as dense dashboards, maps, network diagrams, or highly nested multi-panel figures. The pipeline also uses closed-source MLLMs and image-editing systems during data construction and evaluation; this enables large-scale curation, but it makes independent validation, source-code auditing, and split-integrity checks important for interpreting the results. We will release dataset metadata, task splits, prompts, evaluation code, and trained LoRA adapters, together with documentation of which construction steps depend on proprietary APIs.

## B Qualitative Comparisons

## C Dataset Curation Pipeline Details

This section provides implementation details of the six-stage curation pipeline introduced in Section 3.1 of the main paper. We elaborate on each step's design decisions beyond what space allows in the main text.

### C.1 Step 1: Feasibility Check & Code Generation

Given a source chart image, Gemini-3-Pro (Google, 2025a) assesses feasibility *independently* for each of the four target chart types: table, bar, line, and pie. Each call is governed by a type-specific

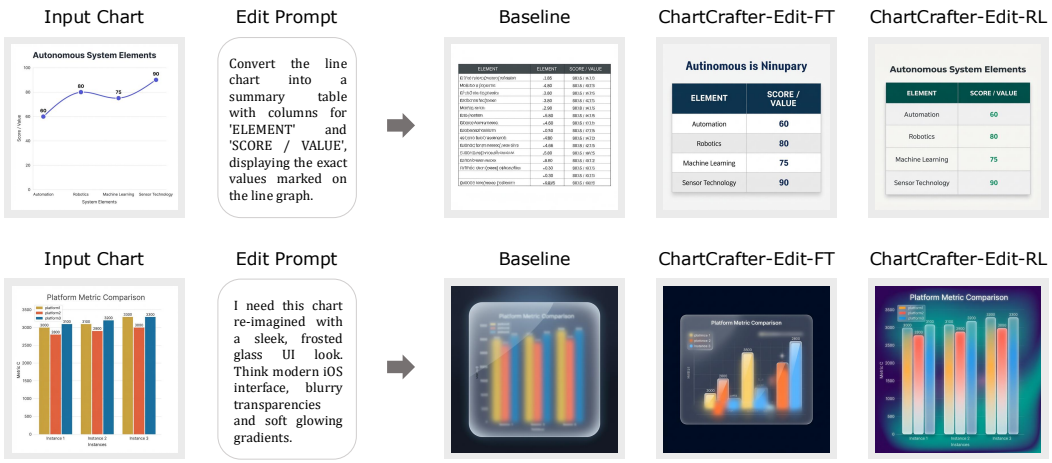


Figure 5: Comparison between the baseline (Qwen-Image-Edit), ChartCrafter-Edit-FT, and -RL, showing progressive aesthetic and contextual improvements.

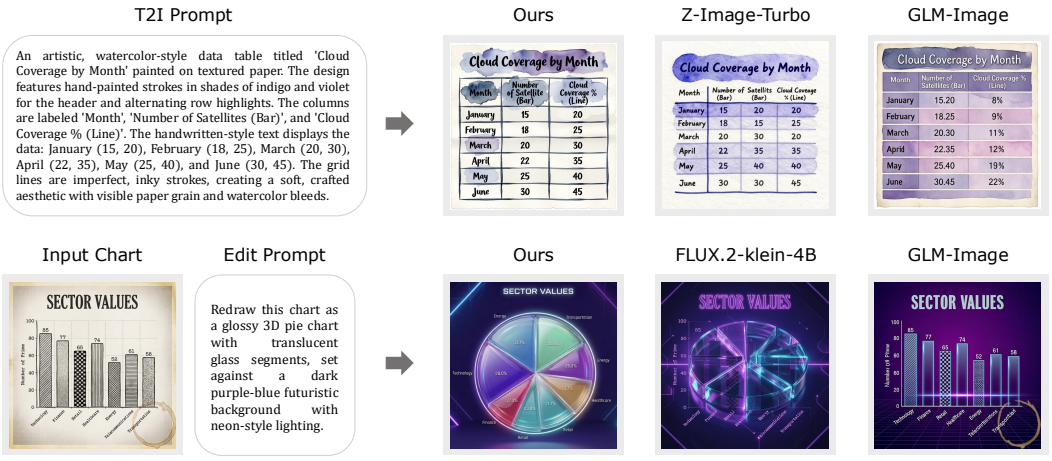


Figure 6: Comparison with open-source image generation models, where ChartCrafter produces charts with better data integrity, readability, and visual coherence.

prompt that instructs the model to determine whether the source image can be meaningfully converted into the target type, and to generate the corresponding rendering code only if feasible. For tables, the code is a complete, self-contained HTML document with embedded CSS. For bar, line, and pie charts, the code is a directly executable Python/Matplotlib (Hunter, 2007) script. A single source image may simultaneously pass feasibility for multiple chart types, each of which proceeds independently through the remaining steps. Infeasible branches are discarded immediately.

### C.2 Step 2: Base Chart Rendering

Feasible rendering code is executed to produce a base rendered chart image:

- **Table (HTML):** The HTML file is loaded in a headless Chromium browser via Playwright and captured as a PNG screenshot.
- **Bar / Line / Pie (Python):** The Matplotlib script is executed directly to produce and save the chart image.

### C.3 STEP 3: AESTHETIC REFINEMENT

Both Steps 3 and 5 call Nano Banana Pro with a 1:1 aspect ratio at 1K resolution. The prompts are shown below.

**Step 3 prompt (chart variant)**

*You are a Senior UI/UX Designer specializing in Information Design. Your task is to upgrade this "raw" chart into a high-fidelity, publication-ready visualization.*
***TRANSFORMATION GOALS:***

1. ***Typography Upgrade**: Replace generic fonts with modern, clean sans-serif typography (like Inter, Roboto, or Helvetica). Ensure hierarchy (Title > Axis Labels > Data).*
2. ***Color Harmony**: Apply a cohesive, professional color palette. Avoid harsh default colors. Use subtle contrasts.*
3. ***De-cluttering**: Remove unnecessary borders or heavy gridlines. Use negative space effectively to let the data breathe.*
4. ***Visual Polish**: Add subtle anti-aliasing and ensure lines/edges are crisp.*

***CRITICAL CONSTRAINT:***

- *PRESERVE all data values, labels, and the general chart structure exactly as is.*
- *Do NOT change the data, only the style.*
- *The goal is "Clean & Modern", not "Artistic & Wild".*

**Step 3 prompt (table variant)**

*This is a DATA TABLE image. Transform it into a more visually appealing and professional table while STRICTLY PRESERVING the table structure.*
***CRITICAL REQUIREMENTS:***

- *Keep it as a TABLE with rows and columns.*
- *Preserve ALL data values exactly as shown.*
- *Maintain the grid structure (horizontal and vertical lines separating cells).*
- *Keep all headers and labels intact.*
- *Do NOT convert it into a chart, graph, or any other visualization.*

***Enhancements allowed:** improve colors and styling, better typography and spacing, modern clean table design, enhanced readability, professional appearance.*
*The output MUST remain a table with the same data structure.*

### C.4 STEP 4: STYLE ANALYSIS & INSTRUCTION SYNTHESIS

Gemini-3-Pro is called with the Step 3 aesthetic chart and the prompt shown below, which embeds the full Visual Aesthetics Mega-Matrix (23 named styles across 5 groups) and 5 user personas. The model returns a JSON object with two fields: `enhancement_instructions` (used as the Step 5 editing prompt), and `short_prompt` (the training editing instruction for same-type Edit tasks).

**Step 4 prompt**

*You are a world-class Art Director and Data Visualization Expert. Transform the provided chart into a STUNNING visual masterpiece.*
***TASK:***

1. ***Analyze the Image**: What is the data about? (Finance? Gaming? Nature? Science?)*
2. ***Select or Invent a Style**: Pick from the **Visual Aesthetics Mega-Matrix** below or invent a contextual style matching the data topic.*
3. ***Select a User Persona**: from Wheel 1 below.*

***— Wheel 1: User Personas —***

- ***Persona A (Executive)**: Brief, professional, demanding. (e.g., "Make it boardroom ready.")*

- ***Persona B (Artist)****: Expressive, poetic, focuses on mood. (e.g., "Give it a dreamy vibe.")*
- ***Persona C (Analyst)****: Precise, technical, focuses on clarity. (e.g., "Increase data-ink ratio.")*
- ***Persona D (Gamer/Geek)****: Internet slang, loves tech culture. (e.g., "Make it look like a HUD.")*
- ***Persona E (Minimalist)****: Hates clutter, wants whitespace. (e.g., "Remove everything unnecessary.")*

***— Wheel 2: Visual Aesthetics Mega-Matrix —***
*[Group 1: Modern & Tech]*

1. ***Cyberpunk****: Neon cyan/magenta lights, dark background, glitch effects, grid overlays.*
2. ***Glassmorphism****: Frosted glass transparency, blur effects, pastel gradients, floating cards.*
3. ***Neomorphism****: Soft shadows, embossed shapes, low contrast, tactile plastic feel.*
4. ***High-Tech HUD****: Thin glowing lines, data widgets, sci-fi blue, transparent interfaces (Iron Man style).*
5. ***Dark Mode UI****: Sleek charcoal grays, high contrast white text, modern app aesthetic.*

*[Group 2: Retro & Nostalgia]*

6. ***80s Synthwave****: Sunset gradients (orange/purple), retro grids, chrome text effects, VHS noise.*
7. ***90s Pixel Art****: Blocky edges, limited color palette, dithering patterns, arcade game look.*
8. ***Vintage Newspaper****: Off-white paper texture, halftone patterns, serif fonts, desaturated ink colors (WSJ style).*
9. ***Blueprint****: Navy blue background, white thin lines, technical annotations, grid paper texture.*
10. ***Pop Art****: Ben-Day dots, bold thick outlines, primary colors (Red/Yellow/Blue), comic book style.*

*[Group 3: Artistic & Organic]*

11. ***Hand-Drawn Sketch****: Pencil or charcoal strokes, rough edges, white paper background, handwritten fonts.*
12. ***Watercolor****: Soft bleeding colors, paper texture, artistic washes, no hard lines.*
13. ***Chalkboard****: Green or black slate texture, white chalk strokes, dusty effects.*
14. ***Oil Painting****: Thick brush strokes, textured canvas, rich blended colors.*
15. ***Collage / Scrapbook****: Ripped paper edges, mixed textures, layered look, cutout aesthetic.*

*[Group 4: Clean & Professional]*

16. ***Swiss International****: Bold typography (Helvetica), asymmetric layout, negative space, red accents.*
17. ***Corporate Flat****: Clean vector shapes, solid brand colors (Salesforce Blue, Slack colors), simple icons.*
18. ***Academic / LaTeX****: Computer Modern font, very thin precise lines, black and white only, formal.*
19. ***Luxury / Editorial****: Gold foil textures, marble background, elegant serif fonts, high-end fashion magazine vibe.*

*[Group 5: Fun & Bold]*

20. ***Memphis Design****: Geometric shapes (squiggles, triangles), confetti patterns, clashing bright colors (80s style).*
21. ***Low Poly****: Faceted geometric shapes, 3D polygonal look, sharp lighting.*
22. ***Claymorphism (3D Cute)****: Soft rounded 3D shapes, plasticine texture, cute lighting, playful colors.*
23. ***Sticker Bomb****: Graffiti style, bold outlines, chaotic energy, street art vibe.*

***Output JSON:*** *{ "enhancement_instructions": technical blueprint with hex codes, textures, lighting effects, "short_prompt": natural language user command matching the chosen persona (3–20 words) }*

### C.5 Step 5: Contextual Augmentation

The `enhancement_instructions` field from Step 4 is passed directly as the editing prompt to Nano Banana Pro (same 1:1 / 1K settings as Step 3).

**Algorithm 1:** ChartCraft pipeline Steps 1–5 (per source image).

**Input:** Source chart image $I_0$; target type set $\mathcal{T} = \{\text{Table, Bar, Line, Pie}\}$
**Output:** For each feasible type $t$: aesthetic image $I_b^t$, contextual image $I_e^t$, enhancement prompt $p_{\text{enh}}^t$, short style prompt $p_{\text{short}}^t$

**foreach** $t \in \mathcal{T}$ **do**
  $\langle feas,\ code\rangle \leftarrow \textsc{Gemini3Pro}(\text{Prompt}_{1,t},\ I_0)$ `// Step 1: feasibility + code generation`
  **if** $feas \neq$ *"YES"* **then skip** $t$
  $I_r^t \leftarrow \textsc{Render}(code,\ t)$ `// Step 2: HTML/Matplotlib rendering`
  $I_b^t \leftarrow \textsc{NanoBananaPro}(\text{Prompt}_{\text{aest}},\ I_r^t)$ `// Step 3: aesthetic refinement`
  $\langle p_{\text{enh}}^t,\ p_{\text{short}}^t\rangle \leftarrow \textsc{Gemini3Pro}(\text{Prompt}_4,\ I_b^t)$ `// Step 4: style analysis & instruction synthesis`
  $I_e^t \leftarrow \textsc{NanoBananaPro}(p_{\text{enh}}^t,\ I_b^t)$ `// Step 5: contextual augmentation`
**end**

Table 13: Source–target image pairs for the four Edit tasks.

| Task | Source | Target |
|---|---|---|
| Chart Aesthetic Refinement | Base rendered ($I_r$) | Aesthetic ($I_b$) |
| Chart Contextualization | Aesthetic ($I_b$) | Contextual ($I_e$) |
| Aesthetic Type Conversion | Aesthetic ($I_b^A$) | Aesthetic ($I_b^B$) |
| Contextual Type Conversion | Contextual ($I_e^A$) | Contextual ($I_e^B$) |

Algorithm 1 summarises Steps 1–5 of the pipeline. Step 6 (post-processing into Edit and T2I training data) is described in Sections D and E.

Sections D and E together describe Step 6 of the pipeline, which post-processes the artifacts from Steps 1–5 into the final training data. For the **Edit** subset, this step pairs pipeline images and attaches editing instructions (reusing Step 4 outputs or generating new ones via Gemini). For the **T2I** subset, this step verifies data consistency across the pipeline outputs and synthesises image-generation prompts.

# D Edit Training Data Construction

The Edit subset (1,390K samples) pairs images produced by the pipeline into four instruction-guided editing tasks, as defined in the main paper. Table 13 lists the source–target image pairs.

## D.1 Same-Type Edit Tasks

Same-type tasks preserve the chart type and underlying data; only the visual style changes between source and target. Their editing instructions are obtained without an additional Gemini call:

- **Chart Aesthetic Refinement** ($I_r \rightarrow I_b$): The editing instruction is randomly sampled from a pre-defined pool of aesthetic refinement prompts (e.g., "Make this chart more visually appealing and professional"). These prompts describe the general beautification goal without specifying a particular style.

- **Chart Contextualization** ($I_b \rightarrow I_e$): The `short_prompt` produced in Step 4 is used directly as the editing instruction. It captures the target style in natural language from one of the five user personas.

### D.2 Cross-Type Edit Tasks

Cross-type tasks change the chart type, so a new editing instruction must be generated. All feasible ordered pairs $(A, B)$ with $A \neq B$ are enumerated, yielding up to six directed pairs per source image (when all four chart types are feasible). Gemini-3-Pro is called with all charts of a given source image simultaneously and asked to evaluate every pair. Pairs where feasibility is judged `NO` are discarded.

**Cross-type editing instruction prompt — Aesthetic variant (used for Aesthetic Type Conversion)**

***# Role***
*You are an expert data annotator for Chart-to-Chart image editing tasks. I will provide {num_types} beautified charts. Types: {types}.*
***# Task***
*Evaluate every possible conversion pair (Source → Target, where Source ≠ Target) and generate a JSON response.*
***# Requirements***

1. ***Feasibility**: Ensure the data content of the two charts is completely identical; otherwise mark "NO".*
2. ***Edit Prompt**: If YES, write a concise instruction starting with an action verb ("Convert", "Transform", "Redraw", "Change"). If NO, leave `edit_prompt` as empty string "".*

***# Output Format***
*Return ONLY a RAW JSON object following the exact schema with all {num_conversion_pairs} pairs.*

**Cross-type editing instruction prompt — Contextual variant (used for Contextual Type Conversion)**

*Same as above, but the input charts are enhanced charts with complex backgrounds (context).*
***Additional requirement for Edit Prompt**: explicitly describe the visual style, background, or medium of the Target image.*

- *If styles differ: describe the new style (e.g., "... into a bar chart sketched on a napkin").*
- *If styles are the same: describe the shared style (e.g., "... maintaining this futuristic dashboard style").*

## E T2I Training Data Construction

The T2I subset (531K samples) is produced by first verifying data consistency across the pipeline outputs and then synthesising image-generation prompts.

### E.1 Consistency Verification

Consistency verification and prompt synthesis are performed in a *single* Gemini-3-Pro call. The model receives the source code from Step 1, the aesthetic chart $I_b$, and the contextual chart $I_e$ simultaneously. If the data across all three inputs is consistent, the model returns two image-generation prompts; otherwise the triple is discarded. The full prompt is shown below.

**T2I consistency verification & prompt synthesis prompt**

***# Role***
*You are an expert data annotator for Chart-to-Image generation tasks.*
***# Task***
*I will provide you with:*

1. ***Source Code**: The code used to generate the chart (contains exact data values).*
2. ***Beautified Image**: The visual output of that code after basic beautification.*
3. ***Enhanced Image**: The final enhanced image with style/context applied.*

*Your task is to:*

Table 14: CraftEval subtasks and applicable evaluation dimensions. ✓ indicates the dimension is evaluated for that subtask; the CSA dimension is only applied where a style instruction is central to the task.

| Split | Subtask | DSI | AQA | CSA |
|---|---|---|---|---|
| Edit | Chart Aesthetic Refinement | ✓ | ✓ | |
| | Chart Contextualization | ✓ | ✓ | ✓ |
| | Aesthetic Chart Type Conversion | ✓ | ✓ | |
| | Contextual Chart Type Conversion | ✓ | ✓ | ✓ |
| T2I | Aesthetic Chart Generation | ✓ | ✓ | |
| | Contextual Chart Generation | ✓ | ✓ | ✓ |

1. ***Check Consistency****: Verify that all three inputs represent the SAME underlying data.*
2. ***Generate Prompts****: If consistent, generate concise prompts for both images.*

***# Source Code***
`{code_content}`
***# Consistency Check Rules***
*The three inputs are consistent if and only if:*

- *The data values (numbers, percentages, labels) in both images match the source code exactly.*
- *The chart type is the same across all three.*
- *The title and axis labels (if any) match.*
- *Minor visual style differences are acceptable, but data must be identical.*

***# Requirements for Prompts (if consistent)***
*Generate **CONCISE** prompts (around 80–120 words) that:*

1. ***Include ALL Key Data****: Title, chart type, axis labels, ALL data points with exact values.*
2. ***Describe Visual Style****: Brief but vivid description of colors, textures, lighting, and aesthetic.*
3. ***Be Precise Yet Compact****: Like a high-quality image prompt, not a detailed specification.*

***# Output Format***
*If NOT consistent:* `{"feasibility": "NO"}`
*If consistent:* `{"feasibility": "YES", "beautified_prompt": "[...]", "enhanced_prompt": "[...]"}`

# F CraftEval Benchmark Construction

## F.1 Subtask Definitions and Dimension Applicability

Table 14 summarises which evaluation dimensions apply to each subtask.

## F.2 Evaluation Prompts

Each dimension is evaluated by Gemini-3-Pro with the Anchor (ground truth) and Test (generated) images, scoring three sub-items on a $\{0, 1, 2\}$ scale. A key design principle is *metric isolation*: each prompt contains a "CRITICAL DIRECTIVE" instructing the model to evaluate *only* its assigned dimension and ignore all others. The full prompts are shown below.

**DSI prompt (applies to all 6 subtasks)**

*You are an expert data visualization evaluator performing a rigorous cross-domain visual comparison. I will provide two images:*

1. ***[Anchor Image]****: The absolute ground truth standard chart.*
2. ***[Test Image]****: The generated chart to be evaluated, which may have complex artistic styles, 3D effects, or background environments.*

*Your task is to evaluate the "Data & Structure Integrity (DSI)" of the [Test Image] using the [Anchor Image] as the absolute visual baseline.*
***CRITICAL DIRECTIVE (STYLE IMMUNITY)**: You MUST completely ignore ALL differences in color palettes, lighting, backgrounds, 3D perspectives, and artistic textures. Your ONLY job is to verify if the underlying physical geometry and text perfectly match the Anchor.*
*Score each sub-item on a strictly* $[0, 1, 2]$ *scale (0=Fail, 1=Partial, 2=Pass):*

1. ***Physical_Framework**: chart type correctness and presence of structural components (axes lines, legend boxes, etc.).*
2. ***Geometric_Topology**: accuracy of data geometry proportions (bar heights, pie slice areas, etc.).*
3. ***Text_Preservation**: completeness and correctness of titles, axis labels, and numerical values.*

*Output ONLY a JSON object with scores and reasoning.*

**AQA prompt (applies to all 6 subtasks)**

*You are an elite graphic designer and UI/UX art director judging a design replication task. I will provide two images:*

1. ***[Anchor Image]**: The absolute ground truth—the "Gold Standard" of perfect aesthetic quality, layout, and color harmony.*
2. ***[Test Image]**: The generated chart to be evaluated.*

***CRITICAL DIRECTIVE (METRIC ISOLATION)**: DO NOT penalize the Test Image for hallucinated text or incorrect data geometries (that is DSI's job). Focus SOLELY on whether the Test Image achieves the same level of visual layout, contrast, clarity, and aesthetic sophistication as the Anchor Image.*
*Score each sub-item on a strictly* $[0, 1, 2]$ *scale:*

1. ***Clarity_Legibility_and_Spacing**: text contrast, margins, absence of visual clutter.*
2. ***Color_Harmony_and_Semantic_Focus**: palette sophistication and visual hierarchy.*
3. ***Layout_Spatial_Balance**: compositional elegance and spatial distribution.*

*Output ONLY a JSON object with scores.*

**CSA prompt (applies to the 3 style-conditioned subtasks; `{user_prompt}` is filled with the editing instruction)**

*You are an expert AI art director evaluating instruction-based visual transformations. I will provide two images and a prompt:*

1. ***[Anchor Image]**: The absolute ground truth—the perfect execution of the user's requested context.*
2. ***[Test Image]**: The generated chart to be evaluated.*

***[User's Context Request]**: {user_prompt}*
***CRITICAL DIRECTIVE (METRIC ISOLATION)**: Look at the Anchor Image as the ultimate reference for HOW the context should look. DO NOT penalize for data structure errors. Focus SOLELY on how closely the Test Image captures the exact thematic vibe, integration quality, and stylistic restraint shown in the Anchor.*
*Score each sub-item on a strictly* $[0, 1, 2]$ *scale:*

1. ***Thematic_Execution**: fidelity to requested aesthetic mood, textures, and atmosphere.*
2. ***Contextual_Cohesion_and_Integration**: seamless fusion of data layer with background, avoiding the "sticker" effect.*
3. ***Readability_and_Restraint**: balance between contextual effects and chart readability.*

*Output ONLY a JSON object with scores and reasoning.*

### F.3 Score Aggregation

Sub-item scores within each dimension are averaged to obtain a per-dimension score in $[0, 2]$, and applicable dimension scores are then averaged and multiplied by 50.




Figure 7: Samples from **Aesthetic Chart Generation** (T2I). Each entry shows a text prompt specifying chart content and a clean, professional visual style, paired with the ground-truth chart image.

## G Dataset Samples

We present representative samples from each of the six ChartCraft tasks. For T2I tasks in Figures 7, 8, 9, and 10, each sample shows the text prompt and the corresponding ground-truth chart. For Edit tasks in Figures 11, 12, 13, 14, 15, and 16, each sample shows the source chart, the natural-language editing instruction, and the ground-truth target chart.




Figure 8: Samples from **Aesthetic Chart Generation** (T2I), continued.




Figure 9: Samples from **Contextual Chart Generation** (T2I). Each entry shows a text prompt describing chart content together with a rich thematic or contextual visual style, paired with the ground-truth chart image.




Figure 10: Samples from **Contextual Chart Generation** (T2I), continued.




Figure 11: Samples from **Chart Aesthetic Refinement** (Edit). Each entry shows the source base-rendered chart, the user styling instruction, and the aesthetically refined ground-truth chart.

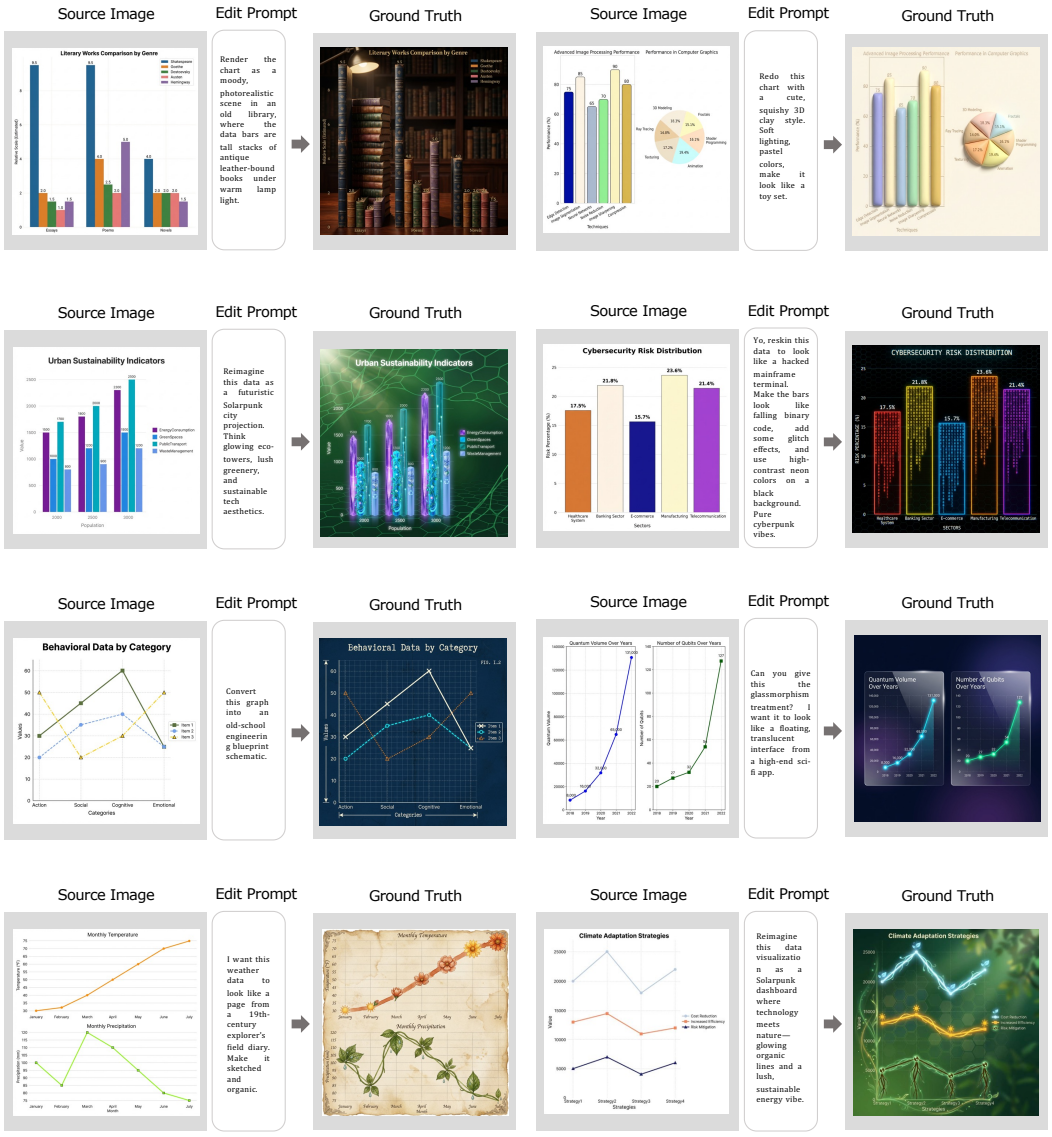


Figure 12: Samples from **Chart Contextualization** (Edit). Each entry shows the source aesthetic chart, the contextual styling instruction, and the context-enriched ground-truth chart.




Figure 13: Samples from **Chart Contextualization** (Edit), continued.

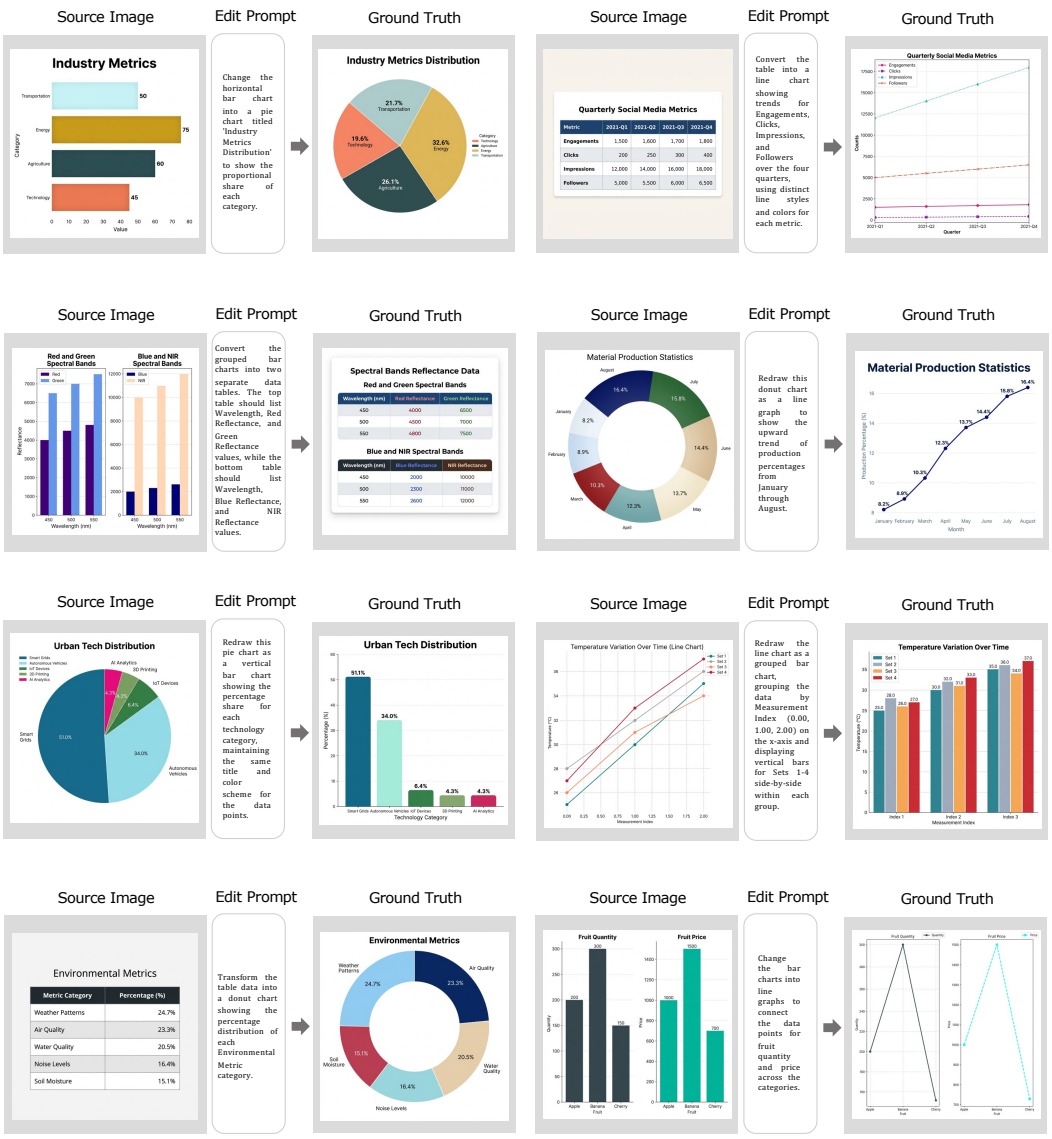


Figure 14: Samples from **Aesthetic Chart Type Conversion** (Edit). Each entry shows the source chart, the type-conversion instruction, and the ground-truth chart of the target type rendered in an aesthetic style.




Figure 15: Samples from **Contextual Chart Type Conversion** (Edit). Each entry shows the source aesthetic chart, the instruction requesting both type change and contextual styling, and the ground-truth target chart.

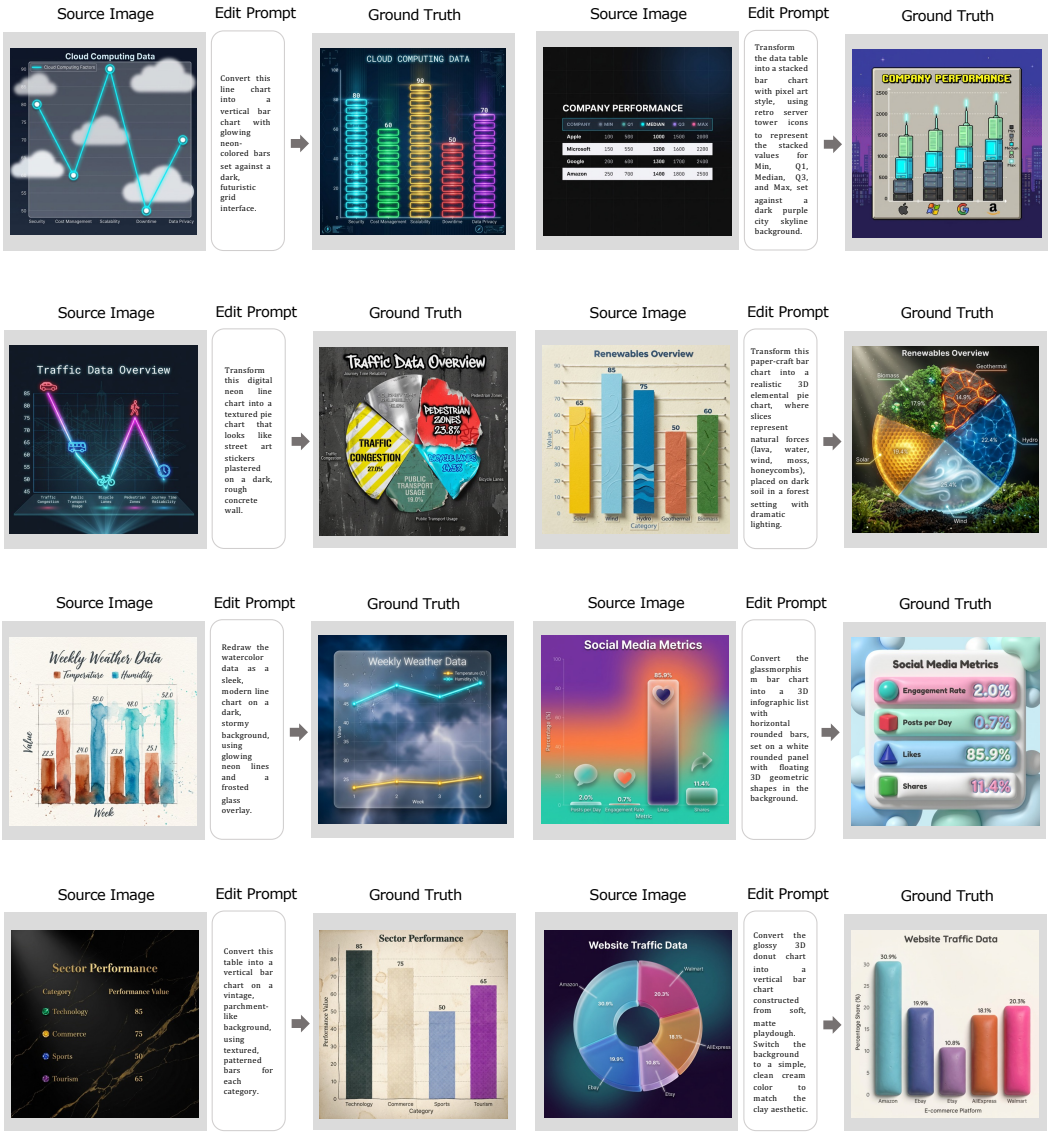


Figure 16: Samples from **Contextual Chart Type Conversion** (Edit), continued.

## H Comparison with Related Resources

Table 15 compares the closest resources by their reported counting units and output representations. Counts of paired edits, unique charts, QA pairs, and evaluation cases measure different quantities; they should not be interpreted as a single scale ranking. Chart-type taxonomies also differ across papers.

Table 15: Resource and task comparison. Sizes describe the cited resources, not the number of examples used in our model training. ChartReformer's original ChartCraft and our resource share a name but are distinct datasets.

| Resource | Reported scale | Representation / task | Relation to this work |
|---|---|---|---|
| ChartReformer / original ChartCraft (Yan et al., 2024) | 109,825 paired edits; line/bar families with five variants | Image + instruction → data table and visual attributes → re-rendered chart | Prior coverage of style, layout, format conversion, and data edits. Our outputs are generated directly in pixel space. |
| ChartEditor / ChartEditVista (Chen et al., 2026) | 7,964 benchmark samples; 31 chart categories | Image + instruction → edited plotting code; SFT and RL | Prior editing and rendering-reward training; our supervision targets aesthetic/contextual image generation and editing. |
| ChartGalaxy (Li et al., 2025) | 1,701,356 synthetic and 61,833 real infographic charts | Charts paired with tables; understanding, code generation, and example-based generation | Prior large-scale infographic design resource; our organization emphasizes six paired generation/editing tasks. |
| IGenBench (Tang et al., 2026) | 600 test cases; 30 infographic types | Text-to-infographic generation; atomic yes/no verification | Prior generation reliability benchmark. CraftEval also includes editing and separate aesthetic/style dimensions. |
| ChartBench (Xu et al., 2023) | Train: 66.6K charts / 599.6K QA; test: 2.1K / 18.9K | Chart reasoning across 9 major categories / 42 subcategories | Tests reading of chart encodings, including unlabeled values; not paired generation/editing supervision. |
| Our ChartCraft / CraftEval | 1,921,633 training samples from 96,336 source images; 600 evaluation cases | Direct image generation/editing; four chart families and six tasks | Paired aesthetic/contextual supervision with separate DSI, AQA, and CSA assessment. |