

CHARTSTYLE-100K: A High-quality Dataset for Structured Visualization Style Transfer

Yuwei Yang^{1†}, Tianchi Xie^{2†}, Jinhong Ni^{1†}, Yukai Guo^{2‡}, Jing Zhang¹,
Liang Zheng¹, Yalong Bai^{3*}, and Yuhui Yuan^{3*}

¹ Australian National University, Australia

² Tsinghua University, China

³ Canva Research, China

{yuwei.yang, jinhong.ni, jing.zhang, liang.zheng}@anu.edu.au,
{xietc24, gyj22}@mails.tsinghua.edu.cn, {josephbai, ryanyuan}@canva.com



Fig. 1: Examples of style transfer of structured images (charts). Col 1: *content* charts. Cols 2 - 7: content charts rendered into different styles. *Style* charts are shown in the red boxes in the bottom row. Within the same row, content is the same. All images in Cols 2-7 are produced by our *ReChart* model: we faithfully preserve the content data and effectively transfer color scheme, typography, etc, from the style chart into the content chart.

Abstract. Given an input image, style transfer aims to recast it into the style of a reference image while preserving its content. While this prob-

* Corresponding author

† Equal contribution. ‡ Work done during an internship at Canva.

Dataset: <https://hf.co/datasets/ChartFoundation/ChartStyle-100k>

Benchmark: <https://hf.co/datasets/ChartFoundation/ChartStyleBench>

Code: <https://github.com/yuweiyang-anu/ChartStyle>

lem is well addressed for natural images, we find that structured images such as charts, diagrams and tables demand capabilities fundamentally distinct from natural image stylization, i.e. visual elements encode data through geometry (bar heights, arc angles, point positions), and dense text must be preserved exactly. We term this *structured style transfer* as changing appearance while keeping strict element-level fidelity. Even frontier models like GPT-Image-1.5 and Nano-Banana-Pro often struggle here, producing inconsistent transfer, structural distortion, and content leakage. The obvious fix, training on data that restyles a content chart toward a reference, inherits these failures: it distorts the content and leaks the reference into the target. We introduce *ChartForge*, a data pipeline that instead builds triplets in reverse: starting from a style image, it produces a target of similar style but different content, then a matching content chart. Because the content is derived from the target, the two stay structurally disentangled from the start, avoiding forward-pipeline leakage. With style-space resampling and multi-dimensional filtering, *ChartForge* yields CHARTSTYLE-100K, over 100K triplets across 36 chart types and 3 visualization families. For evaluation, we further build CHARTSTYLE-BENCH, a benchmark of 300 content-style pairs with 6 metrics. Using CHARTSTYLE-100K, Training *ReChart* by progressively fine-tuning Qwen-Image-Edit on CHARTSTYLE-100K achieves the best overall performance across all baselines, including GPT-Image-1.5 and Nano-Banana-Pro.

Keywords: Structured Visualizations · Style Transfer · Image Editing · Dataset and Benchmark

1 Introduction

Structured visualizations, including charts [25, 26, 60, 62, 63, 65], diagrams [5, 28, 32, 66], flowcharts [13, 45, 67], and tables [52, 68], serve as essential tools for effective data communication, converting quantitative and relational information into intuitive visuals. These visualizations are widely used across news media, business, and education, making complex data comprehensible to broad audiences. Beyond content accuracy, the right visual style transforms a plain chart from merely readable to unmistakably professional, thus shaping how information is perceived, trusted, and retained. However, producing expressive and stylistically coherent visualizations at scale remains labor-intensive, requiring specialized design expertise and domain knowledge.

Motivated by recent breakthroughs in diffusion-based image generation and editing [1, 23, 38, 40, 44, 50, 51, 53, 64], we investigate whether these models can be directly applied to structured visualization style transfer. Despite strong performance on natural images, our evaluation reveals systematic failure patterns even in leading commercial models such as GPT-Image-1.5 [39] and Nano-Banana-Pro [10], as shown in Fig. 2. We attribute these failures to three challenges inherent to structured visualization stylization. First, *Compositional Style*. Chart style is not a single attribute but a compositional ensemble spanning color

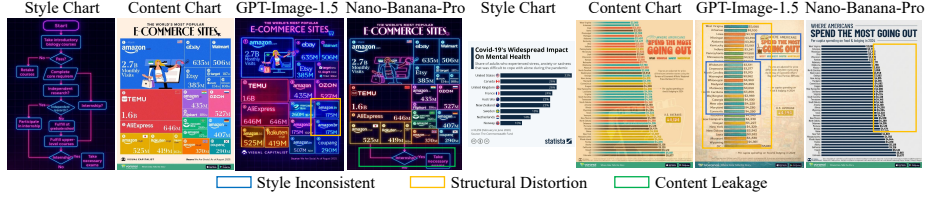


Fig. 2: Illustrating representative failures of latest image editing systems on chart style transfer. Highlighted regions show three error types: style inconsistency, structural distortions that compromise data-encoding geometry, and content leakage.

palettes, typography, mark geometries, annotation layouts, and decorative elements. Faithful transfer thus requires fine-grained control over individual attributes. Second, *Data-encoding Geometry*. Unlike natural images, structured visualizations encode data directly through geometry: bar heights map to values, arc angles to proportions, and point positions to bivariate relationships. Any distortion silently corrupts the data, imposing a zero-tolerance fidelity constraint that perceptually-optimized editors routinely violate. Third, *Style-Content Entanglement*. Style and content share the same visual vocabulary, as I_s and I_c both consist of charts. This overlap makes it hard to separate transferable style from protected content, so layouts, text, or data marks from I_s leak into the output.

Overcoming these challenges requires samples that map a content chart and a style reference to a correctly styled target. Such data does not exist at scale and should be synthesized. However, the natural approach, forward synthesis (restyling a content chart toward a reference), is exactly where existing models fail, yielding unreliable supervision. We therefore build the data in reverse: we first synthesize a stylized target and then derive a structure-aligned content image from it. Since the content is built to match the target, structural fidelity and style-content separation come from the construction rather than the model. This avoids the leakage inherent in forward synthesis.

Building on this idea, we make three contributions. *First*, we realize it as ChartForge, a *reverse-generation* data pipeline, further equipped with style-space resampling and multi-stage LLM+OCR filtering for diversity and quality. *Second*, we curate CHARTSTYLE-100K, a training set of over 100K triplets across 36 chart types and 3 visualization families, together with CHARTSTYLE-BENCH, a 300-pair evaluation benchmark with 6 metrics covering structural fidelity, style consistency, and content integrity. *Third*, we progressively fine-tune Qwen-Image-Edit using CHARTSTYLE-100K to obtain *ReChart*, which performs fine-grained, element-aware transfer and achieves the best overall performance among all baselines, including GPT-Image-1.5 [39] and Nano-Banana-Pro [10], while reducing content leakage.

2 Related Work

Chart synthesis and editing. Chart synthesis and editing constitute fundamental tasks within chart understanding [12, 55, 56, 59]. Recent research on chart

synthesis has explored generating large-scale chart datasets [14, 18, 21, 35, 58, 62, 65] for model training or customized charts [20, 25, 63] for better visualization by leveraging the programmability of libraries such as matplotlib (*e.g.*, ChartMimic [60]) or D3.js (*e.g.*, ChartGalaxy [26]). These synthetic corpora enable the training of models on diverse chart types, but typically focus on content fidelity for visual understanding, while offering limited stylistic control. Chart editing methods [12, 59, 61, 69] further address color, type, text, and layout adjustments, yet remain constrained by predefined code-level heuristics that limit more sophisticated pixel-level edits, such as intricate textures and complex patterns. Beyond charts, similar synthesis and editing paradigms have been extended to other structured visual domains, including diagrams [5, 28, 66], flowcharts [13, 45], and tables [68]. In addition to these studies, some recent works have explored pixel-space approaches [48, 70], such as employing diffusion-based models like Qwen-Image-Edit [53] and FLUX.1 [1] for editing structured visuals. However, these efforts primarily focus on content modification, while the interplay between structure preservation and stylistic variation remains an unresolved challenge for style transfer. This paper tackles these challenges by constructing image triplets that better balance structure preservation and style transfer.

Visual style transfer. Visual style transfer aims to apply the style of a reference image to a target content image or prompt. Early optimization-based methods [8] used Gram matrices to balance content and style losses, later extended by adaptive instance normalization (AdaIN) for universal style transfer [19]. With the advent of large text-to-image (T2I) and image-to-image (I2I) diffusion models, research has shifted toward arbitrary style transfer with minimal tuning. IP-Adapter [64] enables image-conditioned control via lightweight adapters, while InstantStyle [50, 51] achieves style transfer without fine-tuning by decoupling style and content features. Style consistency has been explored through shared-attention mechanisms [16] and Q-Former-based disentanglement in DEADiff [40]. Unified frameworks such as CSGO [57] and StyleStudio [24] expand capabilities to both T2I and I2I settings. However, existing methods primarily target natural image generation and struggle with style transfer for structured visualizations, where maintaining strict structural fidelity while adapting abstract stylistic attributes remains particularly challenging. Our work addresses these challenges by learning structured-visualization style transfer from large-scale paired data.

Image generation and editing model. High-fidelity text-to-image generation lies at the core of modern generative AI, catalyzed by diffusion models [7, 17, 29, 31, 41–43, 46]. Recent advances in Flow Matching [29] and Rectified Flow [31] underpin systems such as Stable Diffusion 3 [47], FLUX.2 [23], and commercial models including GPT-Image-1.5 [39], Nano-Banana-Pro [10], and Seedream [44]. Early editing methods improved controllability through guided denoising in SDEdit [34], training-free text control in Prompt-to-Prompt [15], real-image inversion in Null-Text Inversion [36], and instruction-based conditional diffusion in InstructPix2Pix [2]. FlowEdit [22] then extended editing to flow models with inversion-free ODE paths, and recent editors such as BAGEL [6], Step1X-Edit [30], FLUX.2 [23], and Qwen-Image-Edit [53] are built natively on

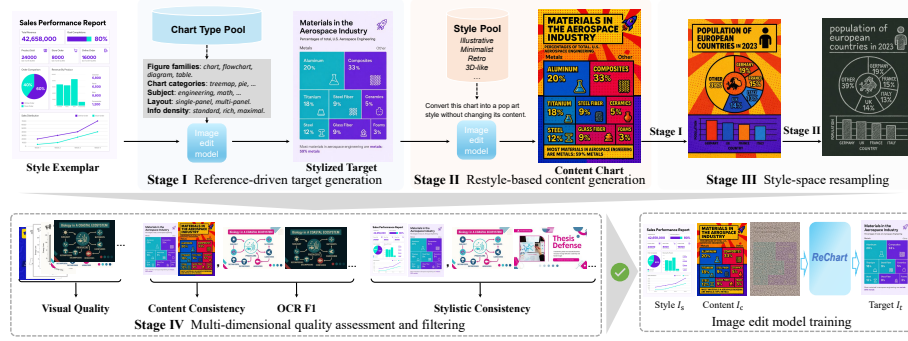


Fig. 3: ChartForge framework. The generation pipeline consists of four stages: (I) Reference-driven target generation, (II) Restyle-based content generation, (III) Style-space resampling, and (IV) Multi-dimensional quality assessment and filtering. These stages jointly produce diverse, structure-consistent, and high-quality image triplets (style I_s , content I_c , target I_t), which are then used to train *ReChart*.

flow matching. Building on these editors, we develop a model closing the gap between general-purpose editing and the content-preservation demands of structured visualization style transfer.

3 Method

3.1 Task Definition

Given a style reference I_s and a content image I_c with geometry $\mathcal{G}_c$ and text $\mathcal{T}_c$, chart style transfer produces a target $I_t = \Phi(I_s, I_c)$ that preserves the structure of I_c while adopting the appearance of I_s . Formally, the task seeks I_t that minimizes:

$$\mathcal{L} = \underbrace{\mathcal{L}_{\text{comp}}(I_t, I_s)}_{\text{compositional style}} + \underbrace{\lambda_s \mathcal{L}_{\text{struct}}(I_t, I_c)}_{\text{geometry preservation}} + \underbrace{\lambda_e \mathcal{L}_{\text{entgl}}(I_t, I_s, I_c)}_{\text{style-content entanglement}},$$

where $\mathcal{L}_{\text{comp}}$ encourages adopting the compositional style of I_s (color, typography, and graphical marks), $\mathcal{L}_{\text{struct}}$ enforces $\|\mathcal{G}(I_t) - \mathcal{G}_c\| \rightarrow 0$ and $\text{OCR}(I_t) = \mathcal{T}_c$, and $\mathcal{L}_{\text{entgl}}$ penalizes content from I_s leaking into I_t . Here, λ_s and λ_e are weighting coefficients that balance the three terms. Since diffusion models treat all tokens uniformly, optimizing $\mathcal{L}_{\text{comp}}$ inevitably perturbs the other two, so the three terms cannot be satisfied by a generic editor alone. We therefore address these challenges at the data level through three complementary principles: *controlled restyling* encourages the separation of content from style via explicit style-family reassignment; *reverse generation* synthesizes the target first to better preserve structural alignment; and *multi-stage quality control* combines LLM-based scoring with OCR verification to suppress leakage at scale. We instantiate these principles in the four-stage pipeline described below, then train our model *ReChart* to balance all three objectives.

Table 1: Dataset statistics of CHARTSTYLE-100K. The dataset spans diverse style sources, subject domains, and style families for restyling, together with large-scale synthesis and quality-controlled filtering that define its final coverage.

CHARTSTYLE-100K: Dataset Summary		
Style sources		
Real infographic charts (from ChartGalaxy)		54,065
Professional designs (from Canva platform)		4,096
<i>Total style references</i>		58,161
Coverage		
Visualization families	Charts / Flowcharts / Diagrams / Tables	
Chart types		36
Subject domains		26
Style families (Restyling)		90
Synthesis scale		
Stage I targets (Charts / FDT)		66,309 / 9,978
Stage II triplets (Charts / FDT)		208,463 / 19,953
Stage III additional triplets		79,993
<i>Candidates before filtering</i>		308,409
Filtering & final set		
Quality signals	$R_{vq}^c \geq 4 / R_{vq}^t \geq 4, R_{cc} \geq 3, R_{sc} \geq 3, R_{ocr} \geq 0.8$	
<i>Final triplets</i>		100,744

3.2 CHARTSTYLE-100K: Dataset Construction

We construct CHARTSTYLE-100K with ChartForge, a data synthesis pipeline built on the reverse-generation principle and organized into two phases. In the *generation phase*, given a style exemplar I_s and a set of randomly sampled content parameters (visualization type, subject, layout, and density), we synthesize a target I_t that matches the appearance of I_s , and then restyle it under a different visual family to obtain its content counterpart I_c : $I_s \xrightarrow{\text{generate}} I_t \xrightarrow{\text{restyle}} I_c$. The *training phase* then learns the reverse mapping $\Phi: (I_s, I_c) \rightarrow I_t$. All synthesis steps use GPT-Image-1 [38] with carefully designed prompts; it serves only as a generator for these subtasks, not as an end-task teacher, so our final model is trained on the resulting triplets rather than distilled from any single model. The pipeline comprises four stages, as illustrated in Fig. 3.

Reference-driven target generation (Stage I). Following the reverse generation principle, we generate the stylized target before any content image, so Stage I needs only a single style exemplar I_s : we synthesize a target I_t whose visual appearance closely follows I_s , while its underlying content is sampled independently rather than taken from any existing chart. Content parameters (visualization type, subject domain, layout, and information density) are drawn from a set of curated pools to ensure broad and balanced coverage (see Tab. 1).

Restyle-based content generation (Stage II). With I_t in hand, we need a structurally identical counterpart rendered in a different style to serve as I_c . Directly stripping style from I_t is ill-defined; instead, we prompt the generator to reproduce its content under a new visual family, sampled from 90 curated aesthetics (*e.g.*, data journalism, minimalist, retro, 3D-like). The prompt explicitly instructs the model to alter only palette, tone, typography, and texture while keeping geometry, text, and layout intact. When multiple variants are generated ($|I_c| > 1$), one is forced to adopt a plain baseline style (matplotlib or hand-drawn

sketch) to anchor the distribution toward common use cases. The result is a clean triplet (I_s, I_c, I_t) in which content and style are explicitly disentangled.

Style-space resampling (Stage III). Real-world style exemplars are dominated by flat-design infographics, leaving artistic and domain-specific aesthetics (*e.g.*, painterly, pixel art, editorial illustration) severely under-represented. To correct this distributional gap, we bootstrap: restyled variants from Stage II are recycled as new style references $\tilde{I}_s$, and the full Stages I–II pipeline is repeated. This self-augmentation expands the effective style space well beyond the original exemplar pool, without requiring additional real-world data collection.

Multi-dimensional quality assessment and filtering (Stage IV). Not all synthesized triplets meet the fidelity requirements for training. We therefore score every triplet (I_s, I_c, I_t) along four complementary dimensions and retain only those that pass all thresholds:

- *Visual quality* $R_{vq} \in [1, 5]$. An LLM-based single-image evaluator scores I_c and I_t independently (R_{vq}^c, R_{vq}^t) on text legibility, contrast, layout discipline, and rendering artifacts.
- *Content consistency* $R_{cc} \in [1, 5]$. The judge jointly evaluates I_c and I_t for semantic alignment: chart structure, axis titles, series correspondence, and textual fidelity.
- *Stylistic consistency* $R_{sc} \in [1, 5]$. It compares I_s and I_t on visual style factors only: color palette, typography, and graphical marks.
- *OCR F1* $R_{ocr} \in [0, 1]$. PaddleOCR [4] extracts text from I_c and I_t ; the token-level F1 quantifies character-level content preservation.

We use GPT-4o [37] as the LLM judge for all quality assessments. The final CHARTSTYLE-100K retains triplets satisfying $R_{vq}^c \geq 4$, $R_{vq}^t \geq 4$, $R_{cc} \geq 3$, $R_{sc} \geq 3$, and $R_{ocr} \geq 0.8$, yielding 100,744 high-quality triplets from approximately 308K candidates. We provide the threshold sensitivity analysis in the supplementary material. Example triplets are shown in Fig. 4.

3.3 CHARTSTYLE-BENCH: Evaluation Benchmark

To systematically evaluate chart style transfer, we construct CHARTSTYLE-BENCH, a benchmark of 300 content–style image pairs.

Content images. We curate 300 content images from three sources: 100 real-world infographic charts from Visual Capitalist, 50 programmatically synthesized charts from ChartGalaxy [26] and ChartMimic [60], and 150 structural visuals (flowcharts, diagrams, tables) collected from Google Images.

Style references. We select 100 style references from Visual Capitalist [49] (disjoint from content images), 100 from InfographicVQA [33], and generate 100 additional structured visuals with diverse styles following Stage II of Sec. 3.2.

Pairing and quality control. Each content image is randomly paired with a style reference, yielding 300 content–style pairs. All images are excluded from CHARTSTYLE-100K to prevent test leakage. To ensure high benchmark quality, all images undergo multi-round manual inspection. We verify structural validity,



Fig. 4: Illustrative examples of CHARTSTYLE-100K Dataset. Each sample (each column) contains a triplet of reference style chart, content chart to be edited, and a target chart used as the ground-truth for training. From left to right, the examples focus on various transfer dimensions, including color scheme transfer, 3D rendering transfer, layout template transfer, typography transfer, and graphical element transfer.

text legibility, and visual clarity, and remove samples with rendering artifacts or ambiguous layouts. We further ensure that content and style images are semantically distinct to avoid trivial transfers. Together, CHARTSTYLE-BENCH covers diverse chart types, style families, and structural complexities, providing a reliable and comprehensive testbed for evaluating chart style transfer methods.

3.4 ReChart: Progressive Fine-Tuning Framework

Although CHARTSTYLE-100K enables direct fine-tuning of Qwen-Image-Edit, a single-stage approach is suboptimal, as the base model has limited exposure to both general style transfer and chart understanding. We therefore adopt progressive fine-tuning through a two-stage curriculum: (i) a first stage on general-domain style transfer data to establish control over fundamental visual elements, followed by (ii) a second stage on CHARTSTYLE-100K for chart-specific specialization. While the first stage substantially improves general style transfer, it leaves performance gaps on charts that the second stage addresses. This progressive strategy yields *ReChart*, enabling robust performance on both general and chart-specific style transfer.

3.5 Structured Transfer Analysis

To better understand and validate the structured transfer behavior learned by *ReChart*, we analyze exemplar-based chart restyling from two complementary views: global token-level influence and local element-level intervention. For the



Fig. 5: Global influence analysis. For each case, the top row shows the style chart, the content chart, and *ReChart*’s target, and the bottom row shows the style→target and content→target influence maps. Style influence concentrates on appearance-sensitive regions, while content influence concentrates on structure-defining regions.

global analysis, we compute each target token’s attention mass to style and content tokens and aggregate it over double-stream MMDiT attention blocks and denoising steps, yielding normalized target-side influence maps:

$$A_s(i) = \frac{1}{Z_i} \text{Avg}_{b,t} \left[\sum_{j \in \text{Style}} \alpha_{i \rightarrow j}^{(b,t)} \right], \quad A_c(i) = \frac{1}{Z_i} \text{Avg}_{b,t} \left[\sum_{j \in \text{Content}} \alpha_{i \rightarrow j}^{(b,t)} \right],$$

where i is a target token, $\alpha_{i \rightarrow j}^{(b,t)}$ denotes attention from target token i to source token j at block b and step t , and Z_i is a per-token normalization term computed over style and content tokens. For the local analysis, we use an element-level intervention to examine the influence of an individual style element. We select an element region R in the style image, defined by a bounding box (*e.g.*, title or legend), remove it, and regenerate the target by *ReChart*. Both the original and intervened targets are generated under an identical setting (same content image, random seed, denoising schedule, and inference hyperparameters), and the per-pixel difference between them, $\Delta_R = |I_t - I_t^{\setminus R}|$, highlights which target regions are most affected by removing R .

Global influence evidence. Fig. 5 shows complementary global roles: style-to-target activation is concentrated on appearance-sensitive regions (texture, tone, rendering pattern), whereas content-to-target activation is concentrated on structure-defining regions (layout and major elements).

Local correspondence. Fig. 6 examines a finer granularity: we remove a single element region (*e.g.*, title or legend) from the style image and regenerate the target image, considering two cases where the corresponding element is either spatially aligned or displaced between the style and target. In both cases, the change in the target tends to appear at regions of the same element type rather

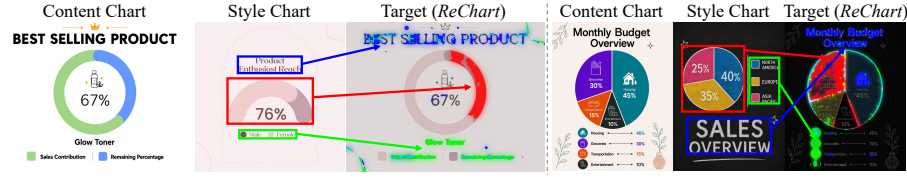


Fig. 6: Local element-level intervention. For each case (left to right: content chart, style chart, target chart), we separately remove individual element regions (e.g., title, legend, or a chart region) from the style chart and overlay the resulting per-pixel changes in *ReChart*’s target. The two cases cover elements that are spatially aligned and spatially displaced between the style and target.

than merely the same spatial location, which may suggest a degree of component-wise sensitivity in the transfer.

Overall observation. The visual evidence above provides qualitative support for addressing the three challenges introduced in Sec. 1. First, for compositional style, style influence is concentrated on appearance-sensitive components, indicating component-aware style injection rather than uniform global stylization. Second, for data-encoding geometry, content influence remains focused on structure-defining regions (layout, major chart elements), consistent with preserving data-carrying geometry during transfer. Third, for style-content entanglement, removing a style element tends to affect target regions of the same element type, suggesting component-wise sensitivity in the transfer.

4 Experiments

4.1 Evaluation Protocols

Metrics. We design a comprehensive set of metrics to evaluate the quality of generated charts across multiple dimensions. For the LLM-as-a-judge evaluation, we employ GPT-4o [37] to assess the following aspects:

- **Content Consistency** measures how well the generated chart preserves the original content of the input chart, rated on a scale of 1 to 5.
- **Style Similarity** assesses how closely the generated chart adheres to the visual style of the reference chart, rated on a scale of 1 to 5.
- **Content Leakage** assesses whether unintended visual elements from the reference chart appear in the generated chart. This metric outputs 1 if content leakage occurs and 0 otherwise.
- **Overall Score** provides a holistic assessment of the generated chart by computing the harmonic mean of content consistency score and style similarity score, where style similarity score is set to 1 if content leakage is detected¹.

¹ In the case of content leakage, the reference chart is leaked into the generated chart, making the style score less meaningful.

Table 2: Quantitative comparisons on CHARTSTYLE-BENCH. $\uparrow$ / $\downarrow$: higher / lower is better. *Severe leakage makes style and fidelity metrics unreliable for models. All existing methods struggle in various aspects. Traditional style transfer methods CSGO [57] and StyleStudio [24] produce significant artifacts. General editing model such as Qwen-Image-Edit [53] has difficulty in fully understanding the instruction, leading to severe leakage. Our *ReChart* achieves competitive performance even against state-of-the-art commercial models including GPT-Image-1.5 [39] and Nano-Banana-Pro [10].

Model	LLM Evaluation*				Others*		
	Content $\uparrow$	Style $\uparrow$	Leakage $\downarrow$	Overall $\uparrow$	Semantic $\uparrow$	Fidelity $\uparrow$	OCRScore $\uparrow$
StyleStudio [24]	1.00	1.67	0.06	1.17	0.58	0.47	0.01
CSGO [57]	1.00	1.97	0.02	1.25	0.55	0.46	0.01
OmniGen2 [54]	1.11	2.66	0.57	1.14	0.54	0.53	0.04
FLUX.2-dev [23]	1.75	3.80	0.80	1.29	0.59	0.73	0.26
Edit-R1 [27]	1.85	3.25	0.63	1.35	0.65	<u>0.66</u>	0.28
Seedream 4.0 [44]	2.39	3.70	0.51	2.00	0.74	0.56	0.50
GPT-Image-1 [38]	2.31	2.96	<u>0.05</u>	2.30	0.77	0.46	0.54
GPT-Image-1.5 [39]	3.30	3.04	0.06	2.82	0.80	0.48	0.73
Nano-Banana [3]	3.57	2.66	0.20	2.44	<u>0.87</u>	0.50	0.77
Nano-Banana-2 [11]	3.48	<u>3.75</u>	0.48	2.59	0.76	0.54	0.71
Nano-Banana-Pro [10]	<u>3.79</u>	3.58	0.32	<u>2.84</u>	0.77	0.54	0.72
Qwen-Image-Edit [53]	2.75	2.73	0.51	1.58	0.76	0.62	0.54
<i>ReChart</i> (Ours)	3.96	2.90	<u>0.05</u>	3.03	0.89	0.49	<u>0.76</u>

We complement the LLM-based judgments with automated perceptual metrics that provide fast, repeatable signals on stylistic and content fidelity. Specifically, we measure the following aspects for every generated chart:

- **Semantic Consistency** compares CLIP embeddings of the content input and generated chart to ensure the semantic structure of the original visualization is preserved despite the style transfer.
- **CLIP Stylistic Fidelity** projects the style reference and generated chart through a CLIP encoder and reports their cosine similarity, capturing high-level alignment in texture, palette, and mark rendering.
- **OCRScore** applies PaddleOCR [4] to both the content and generated charts, computes word-level F1 overlap, and quantifies how faithfully textual annotations and labels are retained.

Baseline Methods. Since structured style transfer takes both a style exemplar and content chart as input, the model must process multi-image inputs. We compare our end-to-end model *ReChart* against: 1) traditional style transfer methods represented by CSGO [57] and StyleStudio [24]; 2) state-of-the-art open-source models including OmniGen2 [54], FLUX.2-dev [23], Edit-R1 [27], Seedream 4.0 [44], Qwen-Image-Edit [53](v2509); and 3) commercial models including GPT-Image-1 [38], Nano-Banana [3], GPT-Image-1.5 [39], Nano-Banana-2 [11] and Nano-Banana-Pro [10].

4.2 Quantitative Results

We present the quantitative results in Tab. 2. The findings reveal that existing image editing models struggle with our complex structured visualization style transfer task. Traditional style transfer methods, such as CSGO [57] and StyleStudio [24], address content leakage by adopting IP-adapter [64], but produce significant artifacts, resulting in poor content fidelity, style adherence, and OCR performance. In contrast, recent open-source editing models, such as FLUX.2-dev [23], Edit-R1 [27], and Qwen-Image-Edit [53], improve content preservation and text rendering due to their increased model capacity. Yet these gains do not resolve the core difficulty. Notably, their style scores are often inflated by copying elements from the style reference. Over half of Qwen-Image-Edit’s samples exhibit content leakage, and FLUX.2-dev leaks severely: its outputs closely resemble the style images, artificially inflating style similarity while degrading overall performance. Even state-of-the-art commercial models, such as GPT-Image-1.5 [39] and Nano-Banana-Pro [10], struggle to fully eliminate content leakage or accurately replicate styles. A detailed per-model leakage analysis is provided in the supplementary material. Our model *ReChart* achieves the best overall performance, leading in content preservation and semantic consistency, while maintaining low content leakage and competitive stylistic fidelity.

4.3 Qualitative Results

We present qualitative results in Fig. 7. Overall, baseline methods often exhibit content leakage and unintended visual artifacts, whereas our approach significantly mitigates these issues while better preserving structural and textual fidelity. We highlight three key observations:

Structural integrity. Preserving the structural integrity of the underlying chart (e.g., precise geometry of the bar) remains a challenge. In our evaluation, GPT-Image-1.5 often perturbs the chart’s global layout—such as altering bar heights/widths or axis spacing—thereby misrepresenting the encoded values (Fig. 7, Row 3, GPT-Image-1.5 changes the overall structure).

Component style adaptation. Complex chart styles (e.g., pixel art) often require substantial visual transformations beyond superficial changes in color or typography. Different elements within a chart (such as icons and background regions) demand distinct stylistic treatments to preserve both readability and structural coherence. Our model effectively performs such differential style adaptation (Fig. 7, Rows 2 and 3, our model changed the style of the man and coins), whereas most baseline models apply a single global transformation.

Text fidelity. Text preservation in current models is scale-dependent. Top-performing methods, including ours and Nano-Banana-Pro, maintain high fidelity for large text but still struggle to preserve the legibility of small-scale annotations. This failure demonstrates that robust, multi-scale text replication remains a significant and open challenge for all editing methods.



Fig. 7: Qualitative comparisons of state-of-the-art models on CHARTSTYLE-BENCH. Existing methods face problems including style misalignment, content inconsistency, text distortion, and content leakage. *ReChart* yields the best overall visual results.

4.4 Ablations

Impact of two-stage training. We ablate the progressive training strategy in Tab. 3. The base Qwen-Image-Edit suffers from severe leakage (0.51). The model trained on general style transfer data reduces leakage to 0.13 and improves content consistency by establishing multi-image alignment. Training solely on chart-specific data further suppresses leakage to 0.03 and improves style understanding, but content consistency drops, indicating that chart data alone is insufficient for fine-grained content preservation. Combining both stages yields the best balance between content preservation and style transfer.

Impact of style-space resampling. Stage III reuses restyled variants as new style references to expand stylistic coverage. Removing it consistently reduces style similarity and overall score (Tab. 4), confirming that real-world exemplars alone under-represent artistic aesthetics and that style-space bootstrapping is necessary for robust generalization.

Table 3: Ablation on the two-stage training, evaluated on CHARTSTYLE-BENCH. Combining both stages yields the best overall performance.

Model	LLM Evaluation				Others		
	Content $\uparrow$	Style $\uparrow$	Leakage $\downarrow$	Overall $\uparrow$	Semantic $\uparrow$	Fidelity $\uparrow$	OCRScore $\uparrow$
Qwen-Image-Edit [53]	2.75	2.73	0.51	1.58	0.76	0.62	0.54
+ stage 1	<u>3.13</u>	<u>2.92</u>	0.13	2.45	0.80	<u>0.52</u>	<u>0.59</u>
+ stage 2	2.45	3.15	0.03	<u>2.53</u>	<u>0.84</u>	0.48	0.54
+ stage 1 + stage 2	3.96	2.90	<u>0.05</u>	3.03	0.89	0.49	0.76

Table 4: Ablation on style-space resampling in data construction, evaluated on CHARTSTYLE-BENCH. Removing it lowers style adherence and overall performance.

Model	LLM Evaluation				Others		
	Content $\uparrow$	Style $\uparrow$	Leakage $\downarrow$	Overall $\uparrow$	Semantic $\uparrow$	Fidelity $\uparrow$	OCR $\uparrow$
<i>ReChart</i>	3.96	2.90	0.05	3.03	0.89	0.49	0.76
w/o style-space resampling	3.57	2.75	0.09	2.71	0.88	0.48	0.73

Table 5: OCR F1 by text size on CHARTSTYLE-BENCH.

Model	<12 px	12–14 px	14–18 px	18–24 px	24–32 px	≥ 32 px
<i>ReChart</i>	0.56	0.56	0.66	0.72	0.75	0.78
GPT-Image-1.5 [39]	0.52	0.58	0.62	0.70	0.70	0.69
Nano-Banana-Pro [10]	0.50	0.60	0.68	0.74	0.71	0.72

Text fidelity across scales. We further analyze text fidelity at different text sizes by grouping text regions by bounding-box height and reporting OCR F1 at each scale (Tab. 5). *ReChart*’s OCR F1 stays stable above 14px ($F1 \geq 0.66$) but drops sharply below it. Strong baselines such as GPT-Image-1.5 and Nano-Banana-Pro show the same trend, confirming that legible small-scale text rendering remains an open challenge for all current models.

Code-based vs. pixel-based style transfer. Structured visualizations are inherently programmatic, and style editing can also be performed through code manipulation in principle. We compare a code-based pipeline with our pixel-based *ReChart* for style transfer. Specifically, we apply an image-to-code model (*i.e.*, Gemini-3-Pro [9]) to convert both the content and style charts into editable code, and then use the same Gemini-3-Pro to modify the content chart’s code to adopt the target font types and visual styles. Fig. 8 presents the resulting visual outputs. Compared to the pixel-based method, the code-based approach struggles to faithfully reproduce complex chart appearances and often introduces rendering artifacts, highlighting the advantages of direct pixel-level editing. We further analyze the trade-offs between pixel- and code-space stylization in the supplementary material.

Reliability of LLM-based evaluation. We use GPT-4o as our primary judge and additionally report Gemini-3-Pro [9] in Tab. 6. Both agree well with humans: across 5 experts and 400 outputs (100 each from Qwen-Image-Edit, GPT-Image-



Fig. 8: Comparison with a code-rendering-based baseline for structured style transfer.

Table 6: Comparison of different LLM-as-judge methods on CHARTSTYLE-BENCH.

Model	LLM Evaluation (GPT-4o / Gemini-3-Pro)			
	Content $\uparrow$	Style $\uparrow$	Leakage $\downarrow$	Overall $\uparrow$
Qwen-Image-Edit [53]	2.75 / 2.57	2.73 / 2.32	0.51 / 0.57	1.58 / 1.35
GPT-Image-1 [38]	2.31 / 2.12	2.96 / 2.78	0.05 / 0.12	2.30 / 2.07
Nano-Banana [3]	3.57 / 3.21	2.66 / 2.22	0.20 / 0.29	2.44 / 2.00
GPT-Image-1.5 [39]	3.30 / 2.55	3.04 / 2.83	0.06 / 0.21	2.82 / 2.28
Nano-Banana-Pro [10]	3.79 / 3.38	3.58 / 3.58	0.32 / 0.44	2.84 / 2.46
ReChart	3.96 / 3.30	2.90 / 2.71	0.05 / 0.15	3.03 / 2.56

1.5, Nano-Banana-Pro, and *ReChart*), GPT-4o and Gemini-3-Pro reach 76.2% / 75.5% pairwise agreement and Spearman $\rho = 0.63/0.74$ (median), 0.52/0.55 (mean). *ReChart* ranks first under both, confirming that our evaluation tracks human judgment and is robust to evaluator choice.

User study. We conduct a user study with 23 participants (10 professional designers, 5 AI researchers, and 8 master’s/PhD students). Each performs double-blind, randomized-order A/B comparisons of *ReChart* against GPT-Image-1.5 and Nano-Banana-Pro, yielding 920 pairwise comparisons per baseline. *ReChart* achieves a 71.9% / 61.1% non-loss rate against GPT-Image-1.5 / Nano-Banana-Pro (53.0% / 43.8% wins), with high per-sample rater agreement (80.2% / 77.8%). These results reflect a clear and consistent human preference for *ReChart*.

5 Conclusion

We study style transfer for structured visualizations and make three contributions. First, *ChartForge*, a reverse-generation pipeline that disentangles style from content at the data level. Second, CHARTSTYLE-100K, a dataset of over 100K quality-filtered triplets across 36 chart types and 3 visualization families, together with CHARTSTYLE-BENCH for standardized evaluation. Third, *ReChart*, an image editing model that achieves the best overall performance while substantially reducing content leakage. Our experiments also reveal limitations common to all current methods, particularly in multi-scale text rendering, indicating that further advances in fine-grained structural control remain an important direction.

References

1. Black Forest Labs: FLUX.1: A new era of AI image generation. <https://github.com/black-forest-labs/flux> (2024), accessed: 2026-06-30
2. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 18392–18402 (2023)
3. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
4. Cui, C., Sun, T., Lin, M., Gao, T., Zhang, Y., Liu, J., Wang, X., Zhang, Z., Zhou, C., Liu, H., Zhang, Y., Lv, W., Huang, K., Zhang, Y., Zhang, J., Zhang, J., Liu, Y., Yu, D., Ma, Y.: Paddleocr 3.0 technical report. arXiv preprint arXiv:2507.05595 (2025)
5. Cui, Z., Yuan, J., Wang, H., Li, Y., Du, C., Ding, Z.: Draw with thought: Unleashing multimodal reasoning for scientific diagram generation. In: Proceedings of the 33rd ACM International Conference on Multimedia (ACM MM). pp. 5050–5059 (2025)
6. Deng, C., Zhu, D., Li, K., Gou, C., Li, F., Wang, Z., Zhong, S., Yu, W., Nie, X., Song, Z., Guang, S., Fan, H.: Emerging properties in unified multimodal pretraining. arXiv preprint arXiv:2505.14683 (2025)
7. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 34, pp. 8780–8794 (2021)
8. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. ArXiv **abs/1508.06576** (2015)
9. Google: Gemini 3 pro: the frontier of vision ai. <https://blog.google/innovation-and-ai/technology/developers-tools/gemini-3-pro-vision/> (2025), accessed: 2026-06-30
10. Google: Introducing nano banana pro. <https://blog.google/innovation-and-ai/products/nano-banana-pro/> (2025), accessed: 2026-06-30
11. Google: Nano banana 2: Combining pro capabilities with lightning-fast speed. <https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/> (2026), accessed: 2026-06-30
12. Han, Y., Zhang, C., Chen, X., Yang, X., Wang, Z., Yu, G., Fu, B., Zhang, H.: Chartllama: A multimodal llm for chart understanding and generation. arXiv preprint arXiv:2311.16483 (2023)
13. He, M., Zeng, J., Jiang, Y., Zhang, W., Liu, Z., Shi, X., Zhou, A.: Flow2code: Evaluating large language models for flowchart-based code generation capability. arXiv preprint arXiv:2506.02073 (2025)
14. He, W., Xi, Z., Zhao, W., Fan, X., Ding, Y., Shan, Z., Gui, T., Zhang, Q., Huang, X.: Distill visual chart reasoning ability from llms to mllms. arXiv preprint arXiv:2410.18798 (2024)
15. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
16. Hertz, A., Voynov, A., Fruchter, S., Cohen-Or, D.: Style aligned image generation via shared attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4775–4785 (2024)

17. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: *Advances in Neural Information Processing Systems (NeurIPS)*. vol. 33, pp. 6840–6851 (2020)
18. Hu, L., Wang, D., Pan, Y., Yu, J., Shao, Y., Feng, C., Nie, L.: Novachart: A large-scale dataset towards chart understanding and generation of multimodal large language models. In: *Proceedings of the 32nd ACM International Conference on Multimedia (ACM MM)*. pp. 3917–3925 (2024)
19. Huang, X., Belongie, S.J.: Arbitrary style transfer in real-time with adaptive instance normalization. *2017 IEEE International Conference on Computer Vision (ICCV)* pp. 1510–1519 (2017)
20. Jain, A., Ramu, P., Garimella, A., Saxena, A.: Doc2chart: Intent-driven zero-shot chart generation from documents. In: *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 34936–34951 (2025)
21. Kondic, J., Li, P., Joshi, D., He, Z., Abedin, S., Sun, J., Wiesel, B., Schwartz, E., Nassar, A., Wu, B., et al.: Chartgen: Scaling chart understanding via code-guided synthetic chart generation. *arXiv preprint arXiv:2507.19492* (2025)
22. Kulikov, E., Gat, O., Matiana, R., Cohen-Or, D., Aberman, K.: FlowEdit: Inversion-free text-based editing using pre-trained flow models. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2025)
23. Labs, B.F.: Flux.2: Analyzing and enhancing the latent space of flux – representation comparison. <https://bfl.ai/research/representation-comparison/> (2025), accessed: 2026-06-30
24. Lei, M., Song, X., Zhu, B., Wang, H., Zhang, C.: StyleStudio: Text-driven style transfer with selective control of style elements. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 23443–23452 (2025)
25. Li, B., Wang, Y., Gu, J., Chang, K.W., Peng, N.: Metal: A multi-agent framework for chart generation with test-time scaling. *arXiv preprint arXiv:2502.17651* (2025)
26. Li, Z., Li, D., Guo, Y., Guo, X., Li, B., Xiao, L., Qiao, S., Chen, J., Wu, Z., Zhang, H., et al.: Chartgalaxy: A dataset for infographic chart understanding and generation. *arXiv preprint arXiv:2505.18668* (2025)
27. Li, Z., Liu, Z., Zhang, Q., Lin, B., Wu, F., Yuan, S., Yan, Z., Ye, Y., Yu, W., Niu, Y., et al.: Uniworld-v2: Reinforce image editing with diffusion negative-aware finetuning and mllm implicit feedback. *arXiv preprint arXiv:2510.16888* (2025)
28. Liang, C., You, J.: Evaluating llm-generated diagrams as graphs. In: *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 12689–12701 (2025)
29. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. *ArXiv abs/2210.02747* (2022)
30. Liu, S., Han, Y., Xing, P., Yin, F., Wang, R., Cheng, W., Liao, J., Wang, Y., Fu, H., Han, C., et al.: Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761* (2025)
31. Liu, X., Wu, C., Liew, T.Y., Wu, J., Zhao, Y., Gong, J., Zhang, R., Li, Z., Wang, Q., Liu, C.: Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003* (2022)
32. Luo, Y., Yuan, Y., Chen, J., Cai, H., Yue, Z., Yang, Y., Dahan, F.Z., Li, J., Lian, Z.: Mmmg: A massive, multidisciplinary, multi-tier generation benchmark for text-to-image reasoning. In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)* (2025)

33. Mathew, M., Bagal, V., Tito, R., Karatzas, D., Valveny, E., Jawahar, C.: Infographicvqa. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1697–1706 (2022)
34. Meng, C., He, Y., Song, Y., Song, J., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (ICLR) (2022)
35. Methani, N., Ganguly, P., Khapra, M.M., Kumar, P.: Plotqa: Reasoning over scientific plots. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 1527–1536 (2020)
36. Mokady, R., Hertz, A., Aberman, K., Pritch, Y., Cohen-Or, D.: Null-text inversion for editing real images using guided diffusion models. arXiv preprint arXiv:2211.09794 (2022)
37. OpenAI: Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/> (2024), accessed: 2026-06-30
38. OpenAI: Introducing 4o image generation. <https://openai.com/index/introducing-4o-image-generation/> (2025), accessed: 2026-06-30
39. OpenAI: The new chatgpt images is here. <https://openai.com/index/new-chatgpt-images-is-here/> (2025), accessed: 2026-06-30
40. Qi, T., Fang, S., Wu, Y., Xie, H., Liu, J., Chen, L., He, Q., Zhang, Y.: Deadiff: An efficient stylization diffusion model with disentangled representations. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 8693–8702 (2024)
41. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with CLIP latents. arXiv preprint arXiv:2204.06125 (2022)
42. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR). pp. 10684–10695 (2022)
43. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K., Ayan, B.K., Mahdavi, S.S., Gontijo Lopes, R., et al.: Photorealistic text-to-image diffusion models with deep language understanding. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 35, pp. 36479–36494 (2022)
44. Seedream, T., Chen, Y., Gao, Y., Gong, L., Guo, M., Guo, Q., Guo, Z., Hou, X., Huang, W., Huang, Y., et al.: Seedream 4.0: Toward next-generation multimodal image generation. arXiv preprint arXiv:2509.20427 (2025)
45. Singh, S., Chaurasia, P., Varun, Y., Pandya, P., Gupta, V., Gupta, V., Roth, D.: Flowvqa: Mapping multimodal logic in visual question answering with flowcharts. In: Findings of the Association for Computational Linguistics ACL 2024. pp. 1330–1350 (2024)
46. Song, Y., Sohl-Dickstein, J.N., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. ArXiv **abs/2011.13456** (2020)
47. Stability AI: Scaling rectified flow transformers for high-resolution image synthesis. <https://stability.ai/news/stable-diffusion-3-research-paper> (2024), accessed: 2026-06-30
48. Tang, Y., Liu, X., Zhang, B., Lan, T., Xie, Y., Lao, J., Wang, Y., Li, H., Gao, T., Pan, B., et al.: Igenbench: Benchmarking the reliability of text-to-infographic generation. arXiv preprint arXiv:2601.04498 (2026)
49. Visual Capitalist (2025), <https://www.visualcapitalist.com>, accessed: 2026-06-30

50. Wang, H., Spinelli, M., Wang, Q., Bai, X., Qin, Z., Chen, A.: Instantstyle: Free lunch towards style-preserving in text-to-image generation. arXiv preprint arXiv:2404.02733 (2024)
51. Wang, H., Xing, P., Huang, R., Ai, H., Wang, Q., Bai, X.: Instantstyle-plus: Style transfer with content-preserving in text-to-image generation. arXiv preprint arXiv:2407.00788 (2024)
52. Wang, Z., Zhang, H., Li, C.L., Eisenschlos, J.M., Perot, V., Wang, Z., Miculicich, L., Fujii, Y., Shang, J., Lee, C.Y., et al.: Chain-of-table: Evolving tables in the reasoning chain for table understanding. In: ICLR (2024)
53. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025)
54. Wu, C., Zheng, P., Yan, R., Xiao, S., Luo, X., Wang, Y., Li, W., Jiang, X., Liu, Y., Zhou, J., et al.: Omnigen2: Exploration to advanced multimodal generation. arXiv preprint arXiv:2506.18871 (2025)
55. Xia, R., Zhang, B., Ye, H., Yan, X., Liu, Q., Zhou, H., Chen, Z., Dou, M., Shi, B., Yan, J., et al.: Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. arXiv preprint arXiv:2402.12185 (2024)
56. Xie, T., Lin, M., Liu, M., Ye, Y., Chen, C., Liu, S.: Infchartqa: A benchmark for multimodal question answering on infographic charts. arXiv preprint arXiv:2505.19028 (2025)
57. Xing, P., Du, K., Wang, Z., Wang, J., Chen, X., Wang, Y., Jin, H., Wu, J., Liu, H., Zhang, Q.: CSGO: Content-style composition in text-to-image generation. In: International Conference on Learning Representations (ICLR) (2025)
58. Xu, Z., Du, S., Qi, Y., Xu, C., Yuan, C., Guo, J.: Chartbench: A benchmark for complex visual reasoning in charts. arXiv preprint arXiv:2312.15915 (2023)
59. Xu, Z., Qu, B., Qi, Y., Du, S., Xu, C., Yuan, C., Guo, J.: Chartmoe: Mixture of diversely aligned expert connector for chart understanding. In: The Thirteenth International Conference on Learning Representations (ICLR) (2025)
60. Yang, C., Shi, C., Liu, Y., Shui, B., Wang, J., Jing, M., Xu, L., Zhu, X., Li, S., Zhang, Y., et al.: Chartmimic: Evaluating llm’s cross-modal reasoning capability via chart-to-code generation. In: International Conference on Learning Representations (ICLR). vol. 2025, pp. 26590–26646 (2025)
61. Yang, D., Zhang, L., Yue, Z., Chen, L., Xu, Y., Wang, W., Jin, Q.: Chartm3: Benchmarking chart editing with multimodal instructions. In: Proceedings of the 33rd ACM International Conference on Multimedia (ACM MM). pp. 5001–5009 (2025)
62. Yang, Y., Zhang, Z., Hou, Y., Li, Z., Liu, G., Payani, A., Ting, Y.S., Zheng, L.: Effective training data synthesis for improving mllm chart understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2653–2663 (2025)
63. Yang, Z., Zhou, Z., Wang, S., Cong, X., Han, X., Yan, Y., Liu, Z., Tan, Z., Liu, P., Yu, D., et al.: Matplotagent: Method and evaluation for llm-based agentic scientific data visualization. In: Findings of the Association for Computational Linguistics ACL 2024. pp. 11789–11804 (2024)
64. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. ArXiv **abs/2308.06721** (2023)
65. Zadeh, F.P., Kim, J., Kim, J.H., Kim, G.: Text2chart3l: Instruction tuning for chart generation with automatic feedback. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 11459–11480 (2024)

66. Zala, A., Lin, H., Cho, J., Bansal, M.: Diagrammergpt: Generating open-domain, open-platform diagrams via llm planning. arXiv preprint arXiv:2310.12128 (2023)
67. Zhang, E., Yao, R., Liu, H., Yu, J., Wang, J.: First multi-dimensional evaluation of flowchart comprehension for multimodal large language models. arXiv preprint arXiv:2406.10057 (2024)
68. Zhang, X., Luo, S., Zhang, B., Ma, Z., Zhang, J., Li, Y., Li, G., Yao, Z., Xu, K., Zhou, J., et al.: Tablellm: Enabling tabular data manipulation by llms in real office usage scenarios. In: Findings of the Association for Computational Linguistics: ACL 2025. pp. 10315–10344 (2025)
69. Zhao, X., Liu, X., Yang, H., Luo, X., Zeng, F., Li, J., Shi, Q., Chen, C.: Chartedit: How far are mllms from automating chart analysis? evaluating mllms’ capability via chart editing. arXiv preprint arXiv:2505.11935 (2025)
70. Zhuo, L., Han, S., Pu, Y., Qiu, B., Paul, S., Liao, Y., Liu, Y., Shao, J., Chen, X., Liu, S., et al.: Factuality matters: When image generation and editing meet structured visuals. arXiv preprint arXiv:2510.05091 (2025)