FONTBENCH: A MULTIDIMENSIONAL BENCHMARK FOR STYLE-CONSISTENT AND CROSS-LINGUAL FONT GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Font generation requires models to jointly preserve character identity, glyph structure, typographic style, and visual quality. However, existing evaluations are often limited to specific scripts, narrow font categories and reconstruction-oriented metrics, making it difficult to assess the comprehensive capabilities of generative models. In this paper, we introduce **FontBench**, a comprehensive benchmark for systematically evaluating font generation across diverse styles, character complexities, and different languages. FontBench contains 920 fonts with three font groups and eight fine-grained subcategories, spanning structurally regular to highly stylized fonts. And FontBench establishes two tasks: style-conditioned and cross-lingual font generation, together with a multidimensional evaluation protocol covering structural fidelity, style consistency, and visual quality. Extensive experiments on five specialized font generation models and seven general-purpose image-to-image models reveal complementary strengths and limitations: general models produce highly readable glyphs but often lack precise structural control, whereas specialized font models better preserve font-specific styles yet struggle with complex artistic fonts. Cross-lingual evaluation further reveals directional asymmetry, with English-to-Chinese limited mainly by structural adaptation and Chinese-to-English constrained primarily by style preservation.

1 INTRODUCTION

Fonts are fundamental visual carriers of written language, determining character readability while conveying stylistic and cultural information in digital communication, graphic design, and human-computer interaction Chen et al. (2014); Wang et al. (2015). With the rapid development of generative models, automatic font generation has achieved remarkable progress, enabling unseen glyph synthesis from limited references through few-shot learning Park et al. (2021a;b), transformer-based architectures Liu & Lian (2023), and diffusion models Yang et al. (2024). Meanwhile, general image-to-image generation and editing models have demonstrated capabilities in visual style imitation Rombach et al. (2022); Radford et al. (2021); Saharia et al. (2022); Ramesh et al. (2022); Xie et al. (2021), providing new possibilities for font synthesis beyond specialized font models. Despite these advances, existing font generation models are evaluated under inconsistent tasks, datasets, and protocols, hindering fair comparison. *To what extent do generative models truly understand fonts?*

To answer this question, comprehensive evaluation of font generation capabilities is essential. Image generation benchmarks, such as T2I-CompBench Huang et al. (2023), GenEval Ghosh et al. (2023), and HEIM Lee et al. (2023), evaluate the quality of generated images from multiple dimensions including compositionality, semantic alignment, robustness, and human preference. However, these benchmarks mainly target natural image generation that cannot capture the unique requirements of font generation. Unlike general images, generated glyphs should simultaneously maintain structural correctness, reproduce reference font styles, and achieve high visual quality.

Existing font generation datasets are mainly designed for specific generation scenarios, as summarized in Table 1. Fonts-100 focuses on few-shot font generation within predefined font categories Li et al. (2020), while LSG-FCST evaluates model generalization through unseen-font and unseen-character splits. TE141K primarily targets artistic text effect transfer rather than general-purpose

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.

Dataset	Fonts	Font Split (Train/Eval)	Char. Split (Train/Eval)	Language	Tasks	Taxonomy	Unseen Font	Complexity Split
Fonts-100 Li et al. (2020)	100	80 / 20	837 / 837	ZH/EN	Font/Effect Transfer	✗	✓	✗
TE141K Yang et al. (2020)	–	–	2,482 / 367	ZH/EN/Others	Text Effect Transfer	✗	✗	✗
FTransGAN Li et al. (2021)	847	818 / 29	972 / 925	ZH/EN	Cross-language Style Transfer	✗	✓	✗
LSG-FCST Li et al. (2024)	820	780 / 29–40	993 / 29–993	ZH	Few-shot Font Generation	✗	✓	✗
DFD Lai et al. (2025)	–	–	–	ZH/EN/Others	Quality Assessment Understanding	4 style categories	✗	✗
FontAnimate Fu et al. (2025)	900	840 / 60	800 / 200–214	ZH	Few-shot Font Generation	✗	✓	✗
SPR Su & Yang (2026)	222	200 / 21	5,000 / 1,000	ZH	Few-shot Font Generation	✗	✓	✗
FontBench	**920**	**524 / 396**	**993 / 225**	**ZH/EN**	**Style-conditioned + Cross-lingual**	**3 groups 8 subcategories**	✓	✓

Table 1: Comparison of FontBench with existing font generation datasets and recent font-related evaluation resources. *Fonts* denotes the number of distinct fonts when explicitly defined by the corresponding resource. *Font Split* and *Char. Split* report the numbers of training and evaluation fonts/characters, respectively; ranges indicate multiple evaluation settings. *ZH* and *EN* denote Chinese and English, respectively. FontBench jointly supports style-conditioned and cross-lingual generation with a hierarchical taxonomy, unseen-font evaluation, and complexity-aware character splits.

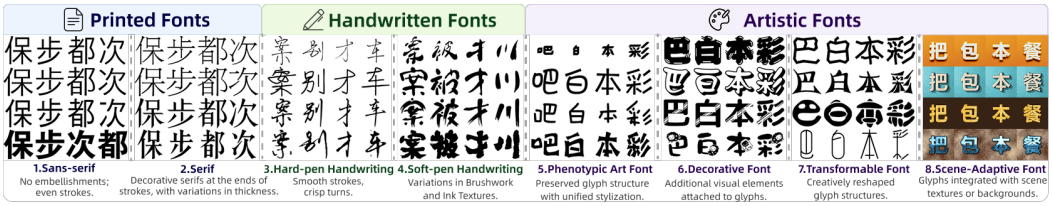


Figure 1: Overview of the font taxonomy adopted in FontBench. Fonts are organized into three major categories, namely *Printed Fonts*, *Handwritten Fonts*, and *Artistic Fonts*, and further divided into eight fine-grained subcategories: *serif, sans-serif, hard-pen handwriting, soft-pen handwriting, phenotypic art, decorative, transformable*, and *scene-adaptive*.

font generation Yang et al. (2020). FTransGAN investigates style transfer between different languages under a specific language setting Li et al. (2021). However, these font generation evaluations are limited to specific character sets, individual writing systems, a narrow range of font styles, relying mainly on reconstruction-oriented metrics such as pixel similarity and perceptual distance. These limitations motivate the development of a unified benchmark that can systematically analyze font generation capabilities across diverse styles, character complexities, and cross-lingual tasks.

In this paper, we introduce **FontBench**, a comprehensive benchmark for evaluating font generation models across diverse font styles, character complexities, and generation tasks, as illustrated in Fig. 1. FontBench organizes 920 fonts into a hierarchical taxonomy based on structural regularity and style complexity. Specifically, fonts are categorized into three groups: **printed**, **handwritten**, and **artistic**, each further divided into fine-grained subcategories. Printed fonts refer to regular typographic styles, including **serif** and **sans-serif** font, which are designed to evaluate the ability of models to accurately reconstruct glyph layouts under structurally constrained conditions. Handwritten fonts refer to human writing styles, including **hard-pen** and **soft-pen** handwriting, aiming to assess whether models can capture stroke dynamics, writing variations, and personalized stylistic characteristics. Artistic fonts contain highly stylized categories, including **decorative**, **transformable**, **phenotypic art**, and **scene-adaptive** font, which evaluate the ability to maintain character identity while handling complex deformations, decorative patterns, and diverse visual effects. Together, this taxonomy provides a progressive evaluation from structurally regular fonts to highly stylized fonts, enabling a comprehensive analysis of font generation capabilities.

FontBench evaluates models through two tasks: **style-conditioned font generation** and **cross-lingual font generation**. Specifically, style-conditioned task measures whether a model can synthesize target characters while preserving the reference font style. Cross-lingual task evaluates whether font styles can be transferred across different languages, including English-to-Chinese and Chinese-to-English. In addition, FontBench establishes a multidimensional evaluation protocol covering **structural fidelity** (OCR-Acc, SSIM, PSNR, and MAE), **style consistency** (CLIP-I and Style-Con), and **visual quality** (LPIPS and FID). Thus, FontBench can identify common failure modes such as unreadable characters, structural distortions, weakened styles, and low-quality artifacts.

Using FontBench, we evaluate five specialized font generation models and seven general image-to-image models. Our experiments reveal several important findings:(1) specialized font generation models generally achieve stronger structural fidelity and style alignment, whereas general image generation models often produce visually plausible but structurally inaccurate glyphs. (2) artistic fonts and complex characters expose substantially larger performance gaps than regular printed fonts. (3) cross-lingual generation exhibits clear directional asymmetry: English-to-Chinese generation is mainly limited by structural adaptation, whereas Chinese-to-English generation is primarily constrained by style preservation. Our contributions are summarized as follows,

• We propose FontBench, a comprehensive font generation benchmark comprising 920 fonts with a hierarchical taxonomy, enabling progressive evaluation from structurally regular fonts to highly stylized fonts.

• We establish two tasks: style-conditioned font generation and cross-lingual font generation, along with a multidimensional protocol: structural fidelity, style consistency, and visual quality.

• We conduct extensive experiments on specialized font models and general-purpose image generation models, offering a comprehensive evaluation of capabilities, strengths and limitations.

2 Related Work

2.1 Font Generation Methods

Font generation aims to synthesize glyphs that preserve character identity while following reference styles Wang et al. (2004; 2023); Hessel et al. (2021). Existing methods mainly disentangle content and style representations to generate unseen glyphs from limited references. Early approaches such as LF-Font, MX-Font, SCFont, and DG-Font explore localized style modeling and structure-aware generation to improve glyph preservation Park et al. (2021a;b); Wang et al. (2004); Shi et al. (2016). Recent transformer- and diffusion-based methods further enhance style representation and visual fidelity Liu & Lian (2023); Hessel et al. (2021); Yao et al. (2024); Yang et al. (2024). Meanwhile, general-purpose image generation models exhibit strong style imitation capabilities, but their ability to maintain glyph identity, structural correctness, and typographic consistency remains underexplored.Existing evaluations are limited in font diversity, scripts, and generation tasks.

2.2 Benchmarking Generative Models

Generative model evaluation has evolved from measuring realism toward multidimensional assessment. Metrics such as FID, LPIPS, and CLIP-based scores evaluate distribution similarity, perceptual quality, and semantic alignment, while benchmarks such as T2I-CompBench, GenEval, and HEIM assess broader capabilities including compositionality, robustness, and human preference Heusel et al. (2017); Zhang et al. (2018); Radford et al. (2021); Huang et al. (2023); Ghosh et al. (2023); Lee et al. (2023). However, these benchmarks mainly focus on natural image generation and cannot fully capture the requirements of font generation. Since generated glyphs must simultaneously preserve readability, structural fidelity, and style consistency Pan et al. (2023); Yao et al. (2024); Shi et al. (2016); Li et al. (2021), a dedicated benchmark covering diverse font styles, scripts, and character complexities is required.

3 Dataset Construction

This section presents the construction of FontBench. We introduce a font taxonomy based on structural regularity and style complexity, describe the split strategy for generalization to unseen font styles, and summarize dataset statistics.

3.1 Font Taxonomy

A key challenge in constructing a font generation benchmark is defining a taxonomy that not only covers diverse font styles but also reflects the different capabilities required for successful font generation. Existing datasets often organize fonts using coarse stylistic labels, making it difficult to

analyze which specific abilities contribute to model success or failure. To address this issue, FontBench categorizes fonts according to two complementary dimensions: *structural regularity* and *style complexity*. Structural regularity characterizes the extent to which glyphs follow canonical character structures, while style complexity reflects the degree of visual variation introduced by stroke deformation, decoration, texture, and artistic effects.Based on these dimensions, we organize fonts into three groups: *printed fonts*, *handwritten fonts*, and *artistic fonts*. These groups form a progressive spectrum from structurally regular fonts to highly stylized fonts, allowing FontBench to systematically evaluate font generation. The taxonomy overview is shown in Fig. 1.

Printed fonts represent the most structurally regular category, including *sans-serif* and *serif* fonts. Sans-serif fonts evaluate a model's ability to preserve overall glyph structure, shape, and spatial layout, serving as the most fundamental and commonly used font category. Serif fonts further assess a model's capability to reproduce fine-grained details, where accurately generating serif terminals is a key indicator of generation precision. Together, these two subcategories evaluate the fundamental capabilities of models to accurately reconstruct glyph layouts under structural constraints.

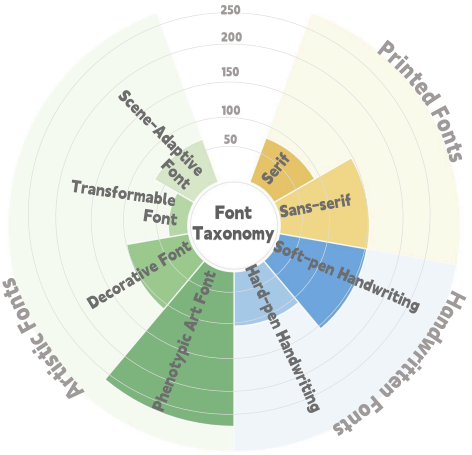


Figure 2: Statistics of the font taxonomy in FontBench. The radial length of each sector indicates the number of fonts contained in the corresponding category.

Handwritten fonts introduce greater stylistic variation while preserving character structures. This category includes *hard-pen handwriting* and *soft-pen handwriting* fonts. Hard-pen fonts emphasize individualized writing habits and local stroke variations, whereas soft-pen fonts further introduce dynamic stroke width, brush effects and irregular trajectories. Consequently, these handwritten fonts evaluate the ability of models to capture stroke dynamics, writing variations, and personalized stylistic characteristics.

Artistic fonts constitute the most challenging category, requiring models to preserve readability while reproducing distinctive visual styles. This category consists of *phenotypic art fonts*, which preserve standard glyph structure and apply a consistent stylistic transformation across the entire font; *transformable fonts*, which create style through substantial deformation of glyph structure; *decorative fonts*, which augment standard glyphs with visual elements, such as ornaments and textures; and *scene-adaptive fonts*, in which glyph appearance is tightly integrated with a particular material, scene, and semantic concept. Therefore, these artistic fonts evaluate robustness to surface stylization, structural deformation, decorative augmentation, and scene-aware typography.

3.2 Split Strategy

FontBench adopts both font-level and character-level splits. In the font-level, FontBench adopts a font-level train/test split to evaluate generalization, where test fonts are completely excluded from the training set to prevent style leakage and ensure evaluation on unseen font styles. Each training font contains 993 characters for learning glyph structures and styles, while each test font contains 225 characters for evaluation. In the character-level, test characters are further divided into **easy** (79), **medium** (131), and **hard** (15) subsets according to stroke count, enabling analysis of model robustness under structural difficulties. This protocol enables systematic evaluation under progressively increasing structural difficulty while maintaining a unified benchmark setting.

3.3 Dataset Statistics

FontBench contains 920 fonts as shown in Table 1 and Fig.2, which were collected from publicly available font libraries and manually screened for completeness and visual quality, including 524 training fonts and 396 test fonts. The test set covers diverse font categories, including 34 hard-pen handwriting fonts, 35 soft-pen handwriting fonts, 40 serif fonts, 58 sans-serif fonts, 50 decorative fonts, 13 transformable fonts, 91 phenotypic art fonts,and 75 scene-adaptive fonts. Fonts are in-

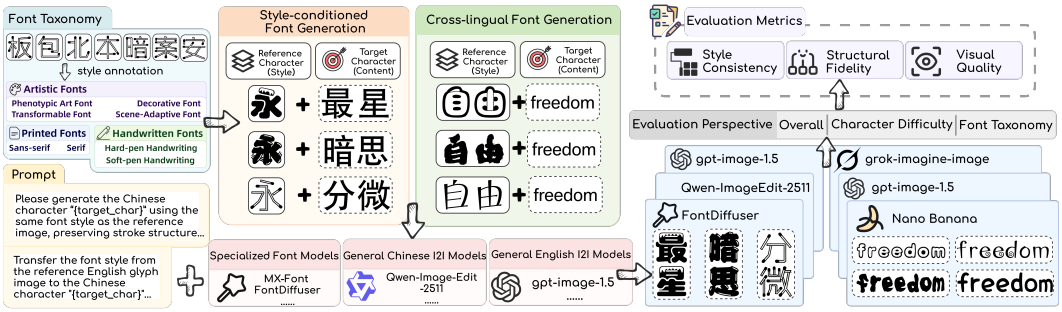


Figure 3: Overview of FontBench, which consists of two evaluation tasks: style-conditioned font generation and cross-lingual font generation. Models take reference glyphs and target character identities as input and generate glyphs in the target style. The outputs are evaluated using a multidimensional protocol, providing a comprehensive assessment of font generation capabilities.

dependently annotated by three annotators who are thoroughly familiar with the FontBench font taxonomy and classification criteria. Disagreements are resolved by majority voting, resulting in a Fleiss' κ of 0.78, indicating substantial inter-annotator agreement.

4 Tasks and Evaluation Protocol

FontBench evaluates font generation from same-language and cross-language perspectives through two tasks: style-conditioned font generation (Task I) and cross-lingual font generation (Task II). As illustrated in Fig. 3, models generate target glyphs from reference glyphs and target character identities under a unified evaluation framework. The generated glyphs are assessed from three dimensions: structural fidelity, style consistency, and visual quality.

4.1 Task I: Style-conditioned Font Generation

Style-conditioned font generation evaluates whether a model can generate unseen target characters while preserving reference glyph styles. Given a reference glyph set $\mathcal{R} = \{r_k\}_{k=1}^{K}$ and a target character t, this task aims to generate a target glyph $\hat{t}$ that satisfies the three criteria,

• **Structural Fidelity**: target glyph $\hat{t}$ preserves the semantic identity and structure of t,

• **Style Consistency**: target glyph $\hat{t}$ remains consistent with the reference font style of $\mathcal{R}$,

• **Visual Quality**: target glyph $\hat{t}$ exhibits high visual quality with clear strokes and minimal artifacts.

Each benchmark sample consists of a reference glyph set, a target character identity, and the corresponding ground-truth glyph. To evaluate the aforementioned criteria, we introduce three evaluation dimensions to compare the generated glyph and the ground-truth glyph (See Appendix B for details):

• For structural fidelity, we introduce **OCR Accuracy**, **SSIM**, **PSNR**, and **MAE**. OCR-Acc measures whether generated glyphs preserve character identity, while SSIM, PSNR, and MAE evaluate structural similarity and pixel-level fidelity between generated and reference glyphs.

• For style consistency, we introduce **CLIP-I** and **Style-Con**. CLIP-I measures visual similarity to reference styles, while Style-Con evaluates intra-font style consistency among generated glyphs.

• For visual quality, we introduce **LPIPS** and **FID**. LPIPS measures perceptual similarity, while FID evaluates distribution similarity between generated and real glyphs.

4.2 Task II: Cross-lingual Font Generation

Cross-lingual font generation evaluates whether a model can transfer font styles across languages. Given a reference glyph set $\mathcal{R} = \{r_k\}_{k=1}^{K}$ from a source language and a target character t_l from another language, the goal of cross-lingual font generation task is to generate a target glyph $\hat{t}_l$ that

Model	Structural Fidelity				Style Consistency		Visual Quality	
	OCR-Acc↑	PSNR↑	SSIM↑	MAE↓	CLIP-I↑	Style-Con↑	LPIPS↓	FID↓
Specialized font generation models								
MX-Font	0.9743	8.7520	0.3916	0.1945	**0.9513**	0.8606	0.1814	22.8427
FUNIT	0.8016	8.0647	0.3363	0.2158	0.9466	0.8669	0.2117	**19.0406**
LF-Font	0.7642	8.9642	0.4109	0.1860	0.9322	0.8533	0.1918	29.6629
FontDiffuser	0.9592	7.9897	0.3472	0.2122	0.9449	0.8545	0.1807	20.4704
IF-Font	0.9423	8.8746	0.4223	0.1848	0.9476	0.8505	0.1439	33.3217
General Chinese image-to-image models								
Qwen-Image-Edit-2511	0.9295	8.3552	0.3880	0.2078	0.8977	0.8586	0.1837	–
FireRed-Image-Edit-1.1	0.9225	8.3939	0.3541	0.2153	0.8774	0.8347	0.2093	–
Doubao-Seedream-4.5	0.9712	6.7687	0.2380	0.2817	0.8800	0.8751	0.2559	–
wan2.6-image	0.9976	7.7280	0.3438	0.2251	0.9115	**0.8945**	0.1806	–
General English image-to-image models								
Gpt-Image-1.5	**1**	10.6140	0.6060	0.1276	0.9114	0.8351	0.1681	–
Grok-Imagine-Image	0.9929	**11.3314**	**0.6398**	**0.1090**	0.9362	0.8601	**0.1349**	–
Nano Banana 2	0.9833	10.8860	0.6039	0.1279	0.9022	0.8266	0.1561	–

Table 2: Quantitative comparison of font generation models across structural fidelity, style consistency, and visual quality. The best results are highlighted in bold, and the second-best results are underlined."–" indicates that FID is unavailable because only a limited number of images were generated for the corresponding models, which is insufficient for reliable FID estimation.

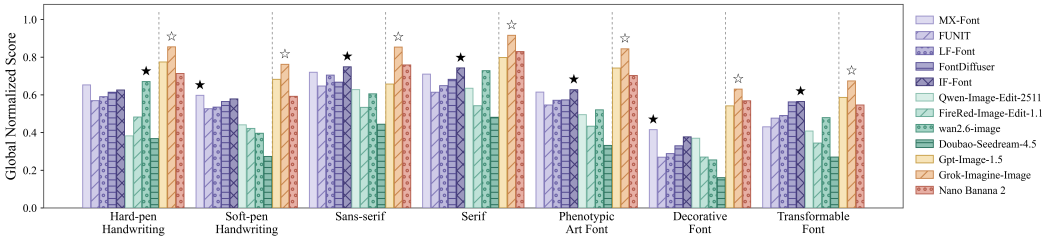


Figure 4: Comparison of global normalized scores across different font styles. Each score is obtained by min–max normalizing multiple evaluation metrics and averaging them across four dimensions. Solid stars indicate the best Chinese-evaluated model within each style, while hollow stars indicate the best English-evaluated model.

satisfies the three criteria: structural fidelity (OCR-Acc, SSIM, PSNR, and MAE), style consistency (CLIP-I and Style-Con) and visual quality (LPIPS and FID). The generated glyph should preserve the semantic identity of the target character, inherit the visual style of the source font, and balance style transfer with structural adaptation.

We construct two representative cross-lingual evaluation settings: **English-to-Chinese** and **Chinese-to-English**. In English-to-Chinese, English glyph images are used as style references, and the model is required to generate Chinese characters in the same visual style. In Chinese-to-English, Chinese glyph images provide the reference style, and the model generates English letters accordingly.

5 Experiment

We evaluate current generation models on FontBench. We first describe the evaluated models, then present quantitative and qualitative analyses of overall performance, character complexity, font categories and cross-lingual font generation.

5.1 Evaluated Models

To understand model capabilities, we evaluate 3 representative families under our unified protocol.

• **Specialized Font Generation Models: MX-Font** Park et al. (2021b), **FUNIT** Liu et al. (2019), **LF-Font** Park et al. (2021a), **FontDiffuser** Yang et al. (2024), and **IF-Font** Chen et al. (2024), which

Model	Structural				Style		Visual
	OCR	PSNR	SSIM	MAE	CLIP-I	Style-Con	LPIPS
Easy Characters							
Specialized font generation models							
MX-Font	0.9369	9.5819	0.4711	0.1609	**0.9509**	0.8865	0.1721
FUNIT	0.7836	8.9209	0.4272	0.1787	0.9473	0.8825	0.1968
LF-Font	0.8813	**9.9565**	**0.4991**	**0.1522**	0.9410	0.8828	**0.1704**
FontDiffuser	0.9025	8.7642	0.4309	0.1766	0.9457	**0.8884**	0.1813
IF-Font	0.8690	9.6938	0.4950	0.1532	0.9470	0.8784	0.1779
General Chinese image-to-image models							
Qwen-Image-Edit-2511	0.8471	8.7946	0.4312	0.1821	0.8891	0.8309	0.1849
FireRed-Image-Edit-1.1	0.8929	9.2948	0.4351	0.1787	0.8773	0.8282	0.1950
Doubao-Seedream-4.5	0.9209	7.4373	0.3639	0.2380	0.8493	0.8285	0.2529
wan2.6-image	**1.0000**	9.0207	0.4905	0.1718	0.9040	0.8658	0.1709
Medium Characters							
Specialized font generation models							
MX-Font	0.9919	8.4688	0.3645	0.2034	**0.9504**	0.8946	0.1807
FUNIT	0.8199	7.7978	0.3063	0.2252	0.9447	0.8914	0.2103
LF-Font	0.8171	**8.7483**	0.3937	0.1912	0.9312	0.8811	0.1891
FontDiffuser	0.9798	7.7227	0.3192	0.2217	0.9428	0.8996	0.1783
IF-Font	0.9715	8.6678	**0.4050**	**0.1907**	0.9476	0.8875	**0.1670**
General Chinese image-to-image models							
Qwen-Image-Edit-2511	0.9657	7.6839	0.3315	0.2330	0.8993	0.8729	0.1826
FireRed-Image-Edit-1.1	0.9184	8.1631	0.3093	0.2275	0.8764	0.8330	0.2035
Doubao-Seedream-4.5	**0.9929**	6.4347	0.1889	0.3017	0.8860	0.9048	0.2561
wan2.6-image	**0.9929**	7.2227	0.2904	0.2027	0.9095	**0.9183**	0.1777
Hard Characters							
Specialized font generation models							
MX-Font	0.9942	8.2051	0.3391	0.2191	**0.9526**	0.9015	0.1916
FUNIT	0.8012	7.4754	0.2673	0.2435	0.9476	0.9002	0.2278
LF-Font	0.5941	7.1879	0.3399	0.2147	0.9244	0.8832	0.2159
FontDiffuser	0.9952	7.4820	0.2914	0.2381	0.9461	**0.9054**	0.1825
IF-Font	0.9863	**8.2621**	**0.3691**	**0.2106**	0.9482	0.8947	**0.1769**
General Chinese image-to-image models							
Qwen-Image-Edit-2511	0.9757	7.6013	0.3229	0.2415	0.9046	0.8630	0.1835
FireRed-Image-Edit-1.1	0.9562	8.0277	0.2885	0.2398	0.8784	0.8429	0.2118
Doubao-Seedream-4.5	**1.0000**	6.4341	0.1613	0.3054	0.9047	0.8919	0.2586
wan2.6-image	**1.0000**	6.9405	0.2505	0.2171	0.9211	0.8993	0.1933

Table 3: Quantitative comparison of font generation models across readability, structural fidelity, style consistency, and visual quality under varying character complexity.

are designed for font synthesis and serve as strong baselines for character preservation, structural fidelity, and style consistency.

• **General Chinese Image-to-image Models: Qwen-Image-Edit-2511** Wu et al. (2025), **FireRed-Image-Edit-1.1** Team et al. (2026), **Doubao-Seedream-4.5** Seedream et al. (2025), and **wan2.6-image**, which are Chinese general-purpose image editing and generation models. Although not designed for font generation, they are evaluated to assess whether general image editing models can handle the structural complexity of Chinese glyphs.

• **General English Image-to-image Models:** GPT-Image-1.5 OpenAI (2025), Grok-Imagine-Image, and Nano-Banana-2, for English and cross-lingual image generation.

5.2 Quantitative results

Overall comparison. Table 2 shows that no single model dominates across all evaluation dimensions, highlighting the necessity of multi-dimensional evaluation for font generation. Specialized font generation models generally achieve more stable style alignment, while general Chinese image-to-image models obtain competitive readability and style consistency. In contrast, general English image-to-image models obtain the best structural fidelity and perceptual quality scores, but these results should be interpreted with caution because English letters are structurally simpler than Chi-

nese characters. Therefore, their strong performance on alphabetic glyphs does not directly imply comparable capability on complex Chinese glyph generation.

Character complexity primarily affects structural fidelity. Table 3 shows the model's performance across different character difficulty levels.As characters become more complex, overall structural fidelity decreases. This trend indicates that complex characters contain more strokes, components, and spatial relationships, making it easier for models to exhibit local structural shifts, missing strokes, or morphological distortion. OCR accuracy is not strictly negatively correlated with character complexity. Another interesting observation is that Style-Con does not decline for Medium and Hard Characters; in fact, it even increases for some models. **This may suggest that complex characters contain more strokes, textures, and local structures, thereby providing richer visual cues for style consistency evaluation.**

The difficulty of font generation is greatly influenced by font style. Fig. 4 shows the results for different font types. Results categorized by type show that the model generally performs better with structurally regular printed font, which feature stable glyphs, clear strokes, and limited decorative variations. In contrast, decorative font, transformable font, and soft-pen handwriting present greater challenges. Decorative font primarily exposes issues related to visual element generation, transformable font expose issues related to structural reshaping, and soft-brush handwriting primarily exposes issues related to dynamic brushstroke modeling—all of which place higher demands on the model's capabilities.

English-to-Chinese generation is substantially more challenging than Chinese-to-English generation. Table 4 reveals a clear directional asymmetry in cross-lingual font generation. Chinese-to-English generation achieves nearly perfect OCR accuracy, indicating that the simpler structures of Latin characters are easier to synthesize. However, CLIP-I and Style-Con remain lower than those of same-language generation, suggesting limited preservation of Chinese font styles. In contrast, English-to-Chinese generation yields substantially lower OCR accuracy, demonstrating that transferring alphabetic font styles to structurally complex Chinese characters is considerably more challenging.

Category	Model	OCR-Acc↑	CLIP-I↑	Style-Con↑
Chinese	Qwen-Image-Edit-2511	0.5904	**0.8766**	**0.7761**
	FireRed-Image-Edit-1.1	**0.6452**	0.8583	0.7581
English	Gpt-Image-1.5	**0.9960**	0.7986	0.6892
	Grok-Imagine-Image	0.9714	0.8093	0.6880
	Nano Banana 2	0.9821	**0.8165**	**0.6897**

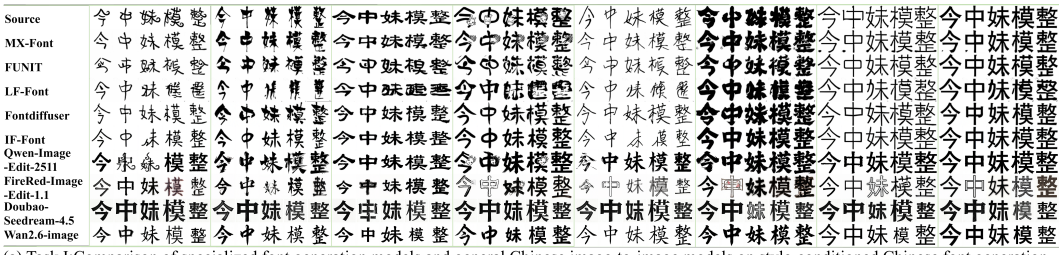
Table 4: Cross-lingual font generation results of representative image editing models.

5.3 Qualitative results

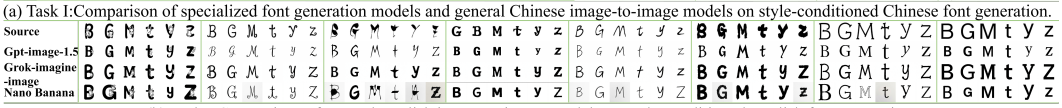
Differences between general-purpose and specialized models. General-purpose models exhibit inconsistent performance; while some characters are reproduced perfectly, others deviate significantly from the reference style. These models sometimes "redraw" characters to make the results appear plausible, yet they do not strictly adhere to the detailed structure of the reference font. In contrast, specialized font generation models typically focus on font style transfer and character structural correspondence, better preserving the weight, contours, and stylistic features of the reference font for certain typefaces. However, when applied to complex artistic or decorative fonts, specialized models may still exhibit issues such as missing local textures, broken strokes, or weakened stylistic elements, as illustrated in Fig. 5(a).

Fonts with complex styles tend to amplify model flaws. As shown in Fig. 5(a-b), the more distinctive a font's stylistic features are, the more pronounced the differences between models become. For example, fonts with bold strokes, decorative textures, handwritten variations, artistic distortions increase the generation difficulty. Some models tend to preserve character structure at the expense of font style, producing results that resemble standard song fonts; others tend to emphasize stylistic effects, but this can lead to chaotic character structures, stroke adhesion, and reduced readability. This indicates a trade-off in font generation between structure preservation and style transfer.

The two transfer directions exhibit distinct failure modes. The qualitative results in Fig. 5(c) and Fig. 5(d) further explain this directional asymmetry. In English-to-Chinese generation, models



(a) Task I:Comparison of specialized font generation models and general Chinese image-to-image models on style-conditioned Chinese font generation.



(b) Task I:Comparison of general English image-to-image models on style-conditioned English font generation.



(c) Task II: Cross-lingual font generation from Chinese to English.



(d) Task II:Cross-lingual font generation from English to Chinese.

Figure 5: Qualitative results on FontBench. The results illustrate notable differences among models in character readability, structural fidelity, and style preservation, particularly for complex artistic fonts and cross-script style transfer scenarios.

often preserve local style cues such as stroke thickness or decorative textures but fail to adapt them to complex Chinese character structures, resulting in component distortion, stroke adhesion, missing components, or source-script leakage, where English letters appear in generated Chinese glyphs. Conversely, Chinese-to-English generation generally produces structurally correct English letters but frequently loses characteristic stylistic elements of the Chinese reference fonts, such as brush dynamics, decorative patterns, and stroke rhythms. These observations indicate that the primary challenge of English-to-Chinese transfer lies in structural adaptation, whereas Chinese-to-English transfer is mainly limited by style abstraction and preservation.

5.4 Human Evaluation

To validate whether the evaluation dimensions of FontBench align with human perception, we conduct a human evaluation on a representative subset of FontBench. Five annotators perform pairwise comparisons of generated glyphs along three dimensions: *Structural Fidelity*, *Style Consistency*, and *Visual Quality*. We compute Spearman rank correlations between the FontBench dimension scores and the corresponding human preference scores.

The results show positive correlations across all three dimensions: $\rho = 0.62$ for Structural Fidelity, $\rho = 0.67$ for Style Consistency, and $\rho = 0.59$ for Visual Quality. These results indicate that the automatic evaluation dimensions are aligned with the corresponding aspects of human perception, providing empirical support for the proposed multi-dimensional evaluation framework. Detailed annotation protocols and implementation details are provided in the supplementary material.

6 Conclusion

We introduced **FontBench**, a comprehensive benchmark for font generation with hierarchical taxonomy, style-conditioned and cross-lingual tasks and multidimensional evaluation metrics. Extensive experiments reveal performance trade-offs across models, font styles, and character complexities, highlighting the need for structure-aware and style-sensitive font generation evaluation .

AI USE STATEMENT

Generative AI tools were used to provide feedback on parts of the experimental and benchmark methodology, assist with the interpretation and presentation of experimental results, and support translation between Chinese and English. They were also used to assist with literature search and summarization, drafting and editing portions of the manuscript, improving readability and organization, and formatting LaTeX text and tables.

Generative AI tools were not used to generate the benchmark dataset, determine ground-truth annotations, formulate mathematical claims or proofs, or autonomously conduct experiments. All AI-assisted content was reviewed and revised by the authors. Methodological suggestions and interpretations were independently assessed against the experimental results, and literature-related information and citations were verified against the corresponding original sources. The authors take full responsibility for the final content of the paper, including all text, claims, experimental results, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the benchmark tasks, dataset splits, evaluation metrics, and experimental protocols in the main paper, with additional information on dataset construction, inference configurations, prompt templates, preprocessing, human evaluation, and fine-grained analyses in the appendix. Due to supplementary-material size limits, the submission includes a representative subset of FontBench. Upon acceptance, we will publicly release the complete FontBench dataset, all benchmark manifests, evaluation scripts, model-specific configurations, and the remaining reproducibility materials. All reported results are produced using fixed manifests and consistent evaluation protocols within each track.

REFERENCES

Guang Chen, Jianchao Yang, Hailin Jin, Jonathan Brandt, Eli Shechtman, Aseem Agarwala, and Tony X Han. Large-scale visual font recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3598–3605, 2014.

Xinping Chen, Xiao Ke, and Wenzhong Guo. If-font: Ideographic description sequence-following font generation. *Advances in Neural Information Processing Systems*, 37:14177–14199, 2024.

Bin Fu, Zixuan Wang, Kainan Yan, Shitian Zhao, Qi Qin, Jie Wen, Junjun He, and Peng Gao. Fontanimate: High quality few-shot font generation via animating font transfer process. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16015–16025. IEEE, 2025.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 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. *Advances in neural information processing systems*, 30, 2017.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023.

Yingxin Lai, Cuijie Xu, Haitian Shi, Guoqing Yang, Xiaoning Li, Zhiming Luo, and Shaozi Li. Fontagent: Enhancing font understanding with large language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19670–19680. IEEE, 2025.

Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Teufel, Marco Bellagente, et al. Holistic evaluation of text-to-image models. *Advances in Neural Information Processing Systems*, 36:69981–70011, 2023.

Chenhao Li, Yuta Taniguchi, Min Lu, and Shin'ichi Konomi. Few-shot font style transfer between different languages. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 433–442, 2021.

Wei Li, Yongxing He, Yanwei Qi, Zejian Li, and Yongchuan Tang. Fet-gan: Font and effect transfer via k-shot adaptive instance normalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 1717–1724, 2020.

Yumei Li, Guangfeng Lin, Menglan He, Dan Yuan, and Kaiyang Liao. Layer similarity guiding few-shot chinese style transfer. *The Visual Computer*, 40(4):2265–2278, 2024.

Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10551–10560, 2019.

Yitian Liu and Zhouhui Lian. Fonttransformer: Few-shot high-resolution chinese glyph image synthesis via stacked transformers. *Pattern Recognition*, 141:109593, 2023.

OpenAI. Introducing gpt-image-1.5: The new chatgpt images. `https://openai.com/index/new-chatgpt-images-is-here/`, December 2025. Accessed: 2026-09-25.

Wei Pan, Anna Zhu, Xinyu Zhou, Brian Kenji Iwana, and Shilin Li. Few shot font generation via transferring similarity guided global style and quantization local style. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19506–19516, 2023.

Seonghyeon Park, Sanghyuk Chun, Junbum Cha, Bado Lee, and Hyunjung Shim. Few-shot font generation with localized style representations and factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 2394–2402, 2021a.

Song Park, Sanghyuk Chun, Junbum Cha, Bado Lee, and Hyunjung Shim. Multiple heads are better than one: Few-shot font generation with multiple localized experts. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 13900–13909, 2021b.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.

Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016.

Peng Su and Xi Yang. Rethinking glyph spatial information in font generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29190–29199, 2026.

Super Intelligence Team, Changhao Qiao, Chao Hui, Chen Li, Cunzheng Wang, Dejia Song, Jiale Zhang, Jing Li, Qiang Xiang, Runqi Wang, et al. Firered-image-edit-1.0 technical report. *arXiv preprint arXiv:2602.13344*, 2026.

Chi Wang, Min Zhou, Tiezheng Ge, Yuning Jiang, Hujun Bao, and Weiwei Xu. Cf-font: Content fusion for few-shot font generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1858–1867, 2023.

Zhangyang Wang, Jianchao Yang, Hailin Jin, Eli Shechtman, Aseem Agarwala, Jonathan Brandt, and Thomas S Huang. Deepfont: Identify your font from an image. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 451–459, 2015.

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

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Yangchen Xie, Xinyuan Chen, Li Sun, and Yue Lu. Dg-font: Deformable generative networks for unsupervised font generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5130–5140, 2021.

Shuai Yang, Wenjing Wang, and Jiaying Liu. Te141k: Artistic text benchmark for text effect transfer. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3709–3723, 2020.

Zhenhua Yang, Dezhi Peng, Yuxin Kong, Yuyi Zhang, Cong Yao, and Lianwen Jin. Fontdiffuser: One-shot font generation via denoising diffusion with multi-scale content aggregation and style contrastive learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 6603–6611, 2024.

Mingshuai Yao, Yabo Zhang, Xianhui Lin, Xiaoming Li, and Wangmeng Zuo. Vq-font: Few-shot font generation with structure-aware enhancement and quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16407–16415, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

A Dataset Details

A.1 Scale and Font-level Split

FontBench contains 920 Chinese fonts and 609,432 rendered glyph images. We use a disjoint font-level split: 524 training fonts provide 993 characters per font (520,332 training glyphs), while 396 test fonts provide 225 characters per font (89,100 test glyphs). No font file or font identity appears in both splits; consequently, all evaluation fonts represent unseen styles. The released split manifests, rather than directory enumeration order, define the benchmark membership and stable sample identifiers.

The 396 test fonts comprise 58 sans-serif, 40 serif, 34 hard-pen handwriting, 35 soft-pen handwriting, 50 decorative, 13 transformable, 91 phenotypic art, and 75 scene-adaptive fonts. These mutually exclusive categories sum to the complete test set and support category-wise analysis without changing the primary model-level aggregation.

A.2 Cross-lingual Style Coverage

In addition to the large-scale Chinese glyph collection, FontBench includes aligned Chinese–English renderings for cross-lingual evaluation. Representative examples are shown in Fig. 6. For the same font style, semantically corresponding Chinese and English content is rendered under a shared visual

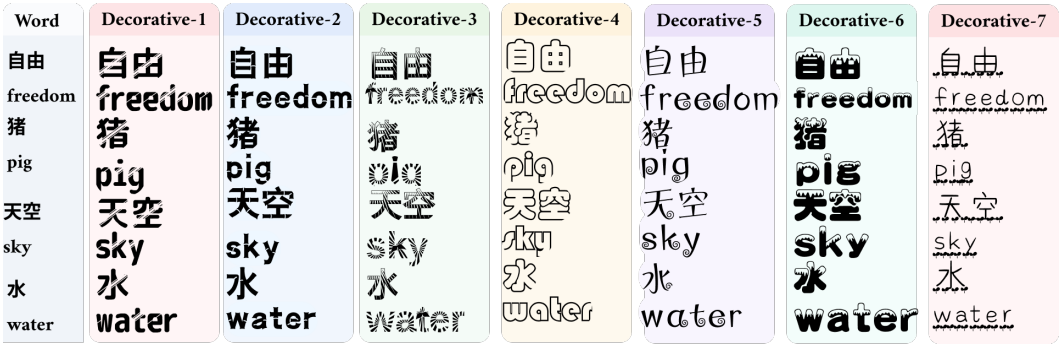


Figure 6: Representative cross-lingual examples in FontBench. Each group shows semantically corresponding Chinese and English content rendered under the same font style, including examples such as "自由" / "freedom", "猪" / "pig", "天空" / "sky", and "水" / "water". The examples illustrate that FontBench covers substantial variation in stroke weight, contour geometry, decoration, and handwritten or artistic appearance across writing systems. Such aligned cross-script examples support controlled evaluation of cross-lingual style transfer rather than evaluating Chinese and English generation independently.

appearance, enabling the benchmark to evaluate whether a model can abstract typographic style from one writing system and transfer it to another.

This cross-script design is particularly important because visual style is not transferred through direct glyph correspondence. Chinese characters and Latin letters differ substantially in structural organization, stroke composition, and spatial layout. A successful cross-lingual font-generation model must therefore preserve high-level style attributes—such as stroke weight, curvature, ornamentation, contour irregularity, and overall visual rhythm—while adapting them to the structural constraints of the target writing system.

The examples in Fig. 6 also show that the difficulty of cross-lingual transfer varies considerably across font styles. Relatively regular styles can often be mapped across scripts through shared stroke-width and geometric characteristics, whereas decorative, handwritten, and highly stylized fonts require the model to preserve more abstract visual properties that do not have direct structural correspondences between Chinese and Latin glyphs. This motivates the bidirectional cross-lingual evaluation in FontBench and complements the style-conditioned generation track.

A.3 Character-complexity Splits

The same fixed set of 225 target characters is used for every test font. Characters are partitioned by the number of writing strokes: **easy** contains 79 characters with at most five strokes, **medium** contains 131 characters with six to eleven strokes, and **hard** contains 15 characters with at least twelve strokes. The three lists are mutually exclusive and exhaustive. The released UTF-8 JSON split files store character-image filenames and are treated as immutable benchmark metadata. Complexity is used only for evaluation and macro-aggregation; it does not alter model inputs or generation settings.

A.4 Evaluation Unit and Reference Glyphs

An evaluation unit is identified by the tuple (font, target character). Its manifest row records the ground-truth image, generated image, font category, character-complexity label, and style-reference images. Style references are selected only from the corresponding real-font directory and never from generated outputs. The same fixed reference identities are supplied to all models within a comparable task whenever their input interfaces permit it. For cross-lingual evaluation, the manifest additionally binds each target word to its source-script reference image, preventing model-specific reference selection.

The specialized Chinese font-generation track uses all 225 test characters. To control proprietary API cost, the general-purpose Chinese image-editing track uses a fixed 45-character subset, the English image-editing track uses a fixed six-character subset, and each cross-lingual direction uses

four fixed target words. These reduced manifests are declared before generation and are shared by every model compared within the corresponding track. Scores from different tracks are reported separately and are not treated as direct like-for-like comparisons.

B Detailed Evaluation Metrics

This section provides the detailed definitions and implementation settings of the evaluation metrics used in FontBench. We evaluate each font separately and report metrics along three dimensions: structural fidelity, style consistency, and visual quality. To avoid the evaluation being dominated by the medium-complexity subset, which contains more characters than the easy- and hard-complexity subsets, we first compute each metric independently on the three character-complexity subsets and then take an unweighted macro-average across the available subsets.

Let $\mathcal{D}_f^c$ denote the set of evaluated target characters for font f and complexity level $c \in \{\text{easy}, \text{medium}, \text{hard}\}$. For a metric m, its score for font f is computed as

$$M_f^{(m)} = \frac{1}{|\mathcal{C}_f^{(m)}|} \sum_{c \in \mathcal{C}_f^{(m)}} \left(\frac{1}{|\mathcal{D}_f^c|} \sum_{i \in \mathcal{D}_f^c} m_i \right), \tag{1}$$

where $\mathcal{C}_f^{(m)}$ contains the complexity subsets for which valid samples are available. This macro-averaging strategy assigns equal weight to the easy, medium, and hard subsets regardless of the number of characters in each subset.

B.1 Structural Fidelity

Structural fidelity measures whether a generated glyph preserves the semantic identity, spatial structure, and pixel-level appearance of its corresponding ground-truth glyph. All pixel-based metrics are computed after converting the images to grayscale, resizing the generated glyph to the spatial resolution of the corresponding ground-truth glyph using bicubic interpolation, and normalizing pixel values to $[0, 1]$.

OCR Accuracy (OCR-Acc). OCR-Acc measures whether a generated glyph can be recognized as the intended target character. We use PaddleOCR with Chinese recognition enabled, text detection disabled, and angle classification disabled. For a generated glyph $\hat{x}_i$ with ground-truth character label y_i, let $h(\hat{x}_i)$ denote the top-ranked OCR prediction after removing whitespace. The OCR score of sample i is

$$m_i^{\text{OCR}} = \mathbf{1}[h(\hat{x}_i) = y_i], \tag{2}$$

where $\mathbf{1}[\cdot]$ is the indicator function. OCR-Acc is first averaged within each complexity subset and then macro-averaged across the easy, medium, and hard subsets according to Eq. equation 1. Although up to three OCR predictions may be returned by the recognition interface, the reported implementation evaluates the top-ranked prediction only; therefore, the metric corresponds to top-1 OCR accuracy.

Mean Absolute Error (MAE). Given a generated grayscale glyph $\hat{x}_i \in [0, 1]^{H \times W}$ and its corresponding ground-truth glyph x_i, MAE is defined as

$$m_i^{\text{MAE}} = \frac{1}{HW} \sum_{p=1}^{H} \sum_{q=1}^{W} \left| x_i^{(p,q)} - \hat{x}_i^{(p,q)} \right|. \tag{3}$$

A lower MAE indicates higher pixel-level fidelity.

Mean Squared Error (MSE). MSE is computed as

$$m_i^{\text{MSE}} = \frac{1}{HW} \sum_{p=1}^{H} \sum_{q=1}^{W} \left(x_i^{(p,q)} - \hat{x}_i^{(p,q)} \right)^2. \tag{4}$$

Although MSE is used internally to calculate PSNR, it may also be reported as an auxiliary reconstruction metric. A lower MSE indicates smaller pixel-level error.

Peak Signal-to-Noise Ratio (PSNR). Because the normalized pixel range is $[0, 1]$, PSNR is defined as

$$m_i^{\mathrm{PSNR}} = 20 \log_{10} \left(\frac{1}{\sqrt{m_i^{\mathrm{MSE}}}} \right). \tag{5}$$

When the MSE is zero, indicating exact pixel-wise reconstruction, the implementation assigns a PSNR value of 100. A higher PSNR indicates better reconstruction fidelity.

Structural Similarity Index Measure (SSIM). SSIM measures structural similarity between the generated and ground-truth grayscale glyphs:

$$m_i^{\mathrm{SSIM}} = \mathrm{SSIM}(x_i, \hat{x}_i), \tag{6}$$

where the dynamic range is fixed to 1.0. SSIM is computed using the implementation provided by `skimage.metrics.structural_similarity`. A higher SSIM indicates greater structural similarity.

B.2 Style Consistency

Style consistency measures whether the generated glyph preserves the visual appearance of the target font. We use the image encoder of OpenCLIP ViT-B/32 pretrained with the OpenAI weights. Each image is converted to RGB and processed using the model-specific OpenCLIP preprocessing pipeline. Let $g(x) \in \mathbb{R}^d$ denote the resulting image embedding after ℓ_2 normalization:

$$\bar{g}(x) = \frac{g(x)}{\|g(x)\|_2}. \tag{7}$$

Cosine similarity between two normalized embeddings is computed by their inner product.

CLIP Image Similarity (CLIP-I). In the reported implementation, CLIP-I measures semantic and visual similarity between each generated glyph and its corresponding ground-truth glyph:

$$m_i^{\text{CLIP-I}} = \bar{g}(\hat{x}_i)^\top \bar{g}(x_i). \tag{8}$$

A higher CLIP-I score indicates that the generated glyph is closer to its ground-truth counterpart in the CLIP image-embedding space. CLIP-I is therefore a paired generated-to-ground-truth similarity metric rather than a generated-to-reference-set metric.

Style Consistency (Style-Con). For each font f, the implementation selects three reference glyphs from the corresponding real-glyph directory. Let the selected reference glyphs be $\mathcal{R}_f = \{r_{f,1}, r_{f,2}, r_{f,3}\}$. Their normalized CLIP embeddings are averaged to form a font-style prototype:

$$s_f = \frac{1}{3} \sum_{k=1}^{3} \bar{g}(r_{f,k}). \tag{9}$$

For a generated glyph $\hat{x}_{f,i}$ belonging to font f, Style-Con is computed as

$$m_{f,i}^{\text{Style-Con}} = \bar{g}(\hat{x}_{f,i})^\top s_f. \tag{10}$$

A higher Style-Con score indicates that the generated glyph is more consistent with the visual style represented by the selected reference glyphs. Following the implementation, the style prototype is the arithmetic mean of the three normalized reference embeddings and is not re-normalized after averaging.

B.3 Visual Quality

Visual quality evaluates whether a generated glyph is perceptually close to its corresponding real glyph and whether it contains visible distortions or artifacts.

Learned Perceptual Image Patch Similarity (LPIPS). LPIPS is computed using the official LPIPS implementation with the AlexNet backbone. The generated and ground-truth glyphs are converted to RGB, resized to the ground-truth resolution using bicubic interpolation, normalized from $[0, 1]$ to $[-1, 1]$, and passed to the pretrained LPIPS network. For each sample,

$$m_i^{\mathrm{LPIPS}} = \mathrm{LPIPS}_{\mathrm{AlexNet}}(x_i, \hat{x}_i). \tag{11}$$

A lower LPIPS value indicates greater perceptual similarity between the generated and ground-truth glyphs.

Fréchet Inception Distance (FID). FID measures distributional similarity between the complete sets of generated and real glyphs:

$$\mathrm{FID} = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}\right), \tag{12}$$

where (μ_r, Σ_r) and (μ_g, Σ_g) are the empirical means and covariance matrices of real and generated glyph features, respectively. We compute FID using `pytorch-fid` v0.3.0 with 2,048-dimensional pool-3 features extracted by its pretrained Inception-v3 network. Input images are converted to RGB, resized to 299×299 using bilinear interpolation, and mapped from $[0, 1]$ to $[-1, 1]$ by the implementation. For each evaluated model, we compute a single FID score over all paired real and generated glyphs in the complete test split using a batch size of 50; FID is not computed separately by font or character-complexity subset.

B.4 Evaluation and Aggregation Details

The expected evaluation set is defined before generation by the released manifest. A prediction is valid only if its sample identifier is unique, its output file exists and is non-empty, and the file can be decoded as an image. For every run, we report the expected sample count, valid prediction count, missing prediction count, failed-request count, duplicate count, and coverage

$$\mathrm{Coverage} = \frac{N_{\mathrm{valid}}}{N_{\mathrm{expected}}}. \tag{13}$$

Extra output files that are not present in the manifest are reported but excluded. A run is eligible for the primary benchmark tables only when coverage is 100%. If an incomplete run is included for diagnostic purposes, it is explicitly labeled as partial, its coverage and denominator are shown, and its scores are not compared with complete runs. Missing predictions are therefore never silently removed to improve a model's score. FID is computed only for complete paired sets.

For a complete run, metrics are first averaged over all target characters within each complexity subset of each font. The easy-, medium-, and hard-subset means are then averaged with equal weights to obtain the final per-font score. Consequently, each complexity subset contributes equally even though the subsets contain different numbers of characters.

The current implementation stores one final score per font for MAE, MSE, PSNR, SSIM, LPIPS, OCR-Acc, CLIP-I, and Style-Con. When reporting a single model-level result in the main tables, the per-font scores should be further averaged over all evaluated fonts:

$$M_{\mathrm{model}}^{(m)} = \frac{1}{F}\sum_{f=1}^{F} M_f^{(m)}, \tag{14}$$

where F denotes the number of valid evaluation fonts.

B.5 Inference Protocol and Configurations

For a controlled comparison, all specialized font-generation models are trained on the same Font-Bench training split using their corresponding public implementations and evaluated on the same held-out test fonts and target characters. At inference time, each model receives one reference glyph per target font and generates exactly one output for each target character, without best-of-N selection.

Table 5: Inference settings for specialized font-generation models. All models use one reference glyph per target font and generate one output for each target character.

Model	Implementation	Style refs.	Output res.	Inference settings
MX-Font	Unified FFG	1	128×128	Deterministic
FUNIT	Unified FFG	1	128×128	Deterministic
LF-Font	Unified FFG	1	128×128	Deterministic
FontDiffuser	Official code	1	96×96	DPM-Solver++; 20 steps; CFG 7.5
IF-Font	Official code	1	128×128	Greedy autoregressive decoding

Table 6: Inference settings for general-purpose image-editing models. All models use one reference image and generate one output for each target character. Provider-managed settings denote sampling parameters that are not exposed by the corresponding API.

Model	Execution	Style refs.	Inference settings
Qwen-Image-Edit-2511	Local	1	20 steps; true CFG 4.0
FireRed-Image-Edit-1.1	Local	1	20 steps
Doubao-Seedream-4.5	API	1	Provider-managed
Wan2.6-Image	API	1	Provider-managed
GPT-Image-1.5	API	1	Provider-managed
Grok-Imagine-Image	API	1	Provider-managed
Nano-Banana-2	API	1	Provider-managed

For general-purpose image-editing models, we use the same reference budget and the same task-specific prompt template across models within each evaluation setting. Each model receives one reference image and generates one output for each target character. For API-based models, parameters that are not exposed by the corresponding service are left to the provider-managed settings.

C Prompt Templates

All general-purpose image editing models are evaluated using a unified prompt template. For a target character t, the prompt is

> apply the font style in this image to the chinese character "t", keep the original stroke thickness, structure and texture, generate only the single character.

The same prompt is used for all fonts and target characters, with only the target character identity substituted accordingly.

Cross-lingual prompt. For the cross-lingual source-to-English task, let w_{en} denote the ground-truth English translation in lowercase letters. We use the following prompt:

> Translate the characters in the given image into their corresponding English words (lowercase letters), while maintaining the font style.
>
> The translation MUST be "w_{en}".
>
> Requirements:
>
> - Output ONLY the word "w_{en}".
> - Do not add extra words or symbols.
> - Keep the font style exactly the same.
> - Use black text on a white background.
> - Use a centered composition.

The value of w_{en} is replaced by the predefined target translation for each manifest item. All evaluated models receive the same prompt and target word for that item.

D HUMAN EVALUATION DETAILS

This section provides additional details on the human evaluation protocol, including sample selection, annotation criteria, and correlation analysis. The human study is designed to examine whether the three evaluation dimensions in FontBench—*Structural Fidelity*, *Style Consistency*, and *Visual Quality*—are aligned with human perception.

D.1 SAMPLE SELECTION

To reduce annotation cost while maintaining coverage across different font styles and character complexities, we construct a fixed evaluation subset from the FontBench test set. We select 30 fonts covering printed, handwritten, and artistic font categories. For each selected font, 20 target characters are sampled across different character-complexity levels, resulting in 600 target instances.

For each target instance, annotators are presented with: (1) the reference glyphs representing the target font style, (2) the target character identity, and (3) two generated glyphs produced by different models. The two generated glyphs are randomly assigned to the left and right positions to reduce positional bias.

D.2 ANNOTATION INSTRUCTIONS

Five annotators independently evaluate the generated glyphs using pairwise comparisons along three criteria.

Structural Fidelity. Annotators are asked: "Which generated glyph better preserves the identity and structural characteristics of the target character?" They are instructed to consider character recognizability, preservation of major components, correctness of stroke arrangement, and structural distortions.

Style Consistency. Annotators are asked: "Which generated glyph better matches the font style represented by the reference glyphs?" They are instructed to consider stroke characteristics, overall visual style, decorative patterns, texture, and other stylistic details.

Visual Quality. Annotators are asked: "Which generated glyph has better visual quality?" They are instructed to consider visible generation artifacts, incomplete strokes, unnatural distortions, and overall visual coherence.

For each criterion, annotators choose one of three options: *Left*, *Right*, or *Tie*. All annotators evaluate the same comparison instances independently.

D.3 CORRELATION ANALYSIS

To evaluate the agreement between the FontBench evaluation dimensions and human perception, the pairwise annotations from the five annotators are aggregated into human preference scores for each evaluation dimension. We then compute Spearman rank correlation between the automatic FontBench dimension scores and the corresponding human preference scores.

Table 7 reports the resulting dimension-level correlations. Structural Fidelity achieves a Spearman correlation of $\rho = 0.62$ with human structural judgments, Style Consistency achieves $\rho = 0.67$ with human style judgments, and Visual Quality achieves $\rho = 0.59$ with human perceptual-quality judgments. All three correlations are positive, indicating consistent alignment between the automatic evaluation dimensions and their corresponding human criteria.

E ADDITIONAL QUANTITATIVE RESULTS

E.1 CATEGORY-WISE AGGREGATE PERFORMANCE

To provide a compact view of model robustness across different font categories, we construct a normalized aggregate score for visualization. The aggregate score is used only for category-wise

Table 7: Spearman rank correlation between FontBench evaluation dimensions and their corresponding human judgments.

Evaluation dimension	**Spearman ρ**
Structural Fidelity	0.62
Style Consistency	0.67
Visual Quality	0.59

analysis and is *not* treated as an additional benchmark metric; all primary comparisons in FontBench are based on the original evaluation metrics.

Because the Chinese- and English-evaluated models belong to different evaluation tracks, normalization is performed independently within each track. For every metric k, the minimum and maximum are computed over all model–category pairs in the corresponding track. For a higher-is-better metric, the normalized value is

$$\tilde{x}^{(k)} = \frac{x^{(k)} - x^{(k)}_{\min}}{x^{(k)}_{\max} - x^{(k)}_{\min}}, \tag{15}$$

whereas for a lower-is-better metric it is

$$\tilde{x}^{(k)} = \frac{x^{(k)}_{\max} - x^{(k)}}{x^{(k)}_{\max} - x^{(k)}_{\min}}. \tag{16}$$

Here, OCR-Acc, PSNR, SSIM, CLIP-I, and Style-Con are treated as higher-is-better metrics, while MAE and LPIPS are treated as lower-is-better metrics.

Directly averaging all normalized metrics would implicitly assign larger weights to evaluation dimensions containing more metrics. We therefore first aggregate metrics within each of the three FontBench evaluation dimensions. Structural Fidelity is computed as

$$S_{\text{str}} = \frac{1}{4}\left(\widetilde{\text{OCR-Acc}} + \widetilde{\text{PSNR}} + \widetilde{\text{SSIM}} + \widetilde{\text{MAE}}\right), \tag{17}$$

Style Consistency is computed as

$$S_{\text{style}} = \frac{1}{2}\left(\widetilde{\text{CLIP-I}} + \widetilde{\text{Style-Con}}\right), \tag{18}$$

and Visual Quality is represented by normalized LPIPS,

$$S_{\text{qual}} = \widetilde{\text{LPIPS}}. \tag{19}$$

The final category-wise visualization score is obtained by equally averaging the three dimensions:

$$S_{\text{agg}} = \frac{S_{\text{str}} + S_{\text{style}} + S_{\text{qual}}}{3}. \tag{20}$$

FID is excluded from this aggregation because it is a distribution-level metric and is not computed for individual model–font-category pairs. This formulation ensures that Structural Fidelity, Style Consistency, and Visual Quality contribute equally to the visualization, regardless of the number of metrics contained in each dimension.

As shown in Fig. 7(a), the Chinese-evaluated models exhibit substantial variation across font categories. Specialized font-generation models generally maintain relatively stable aggregate performance across the evaluated styles, particularly on sans-serif, serif, hard-pen handwriting, phenotypic

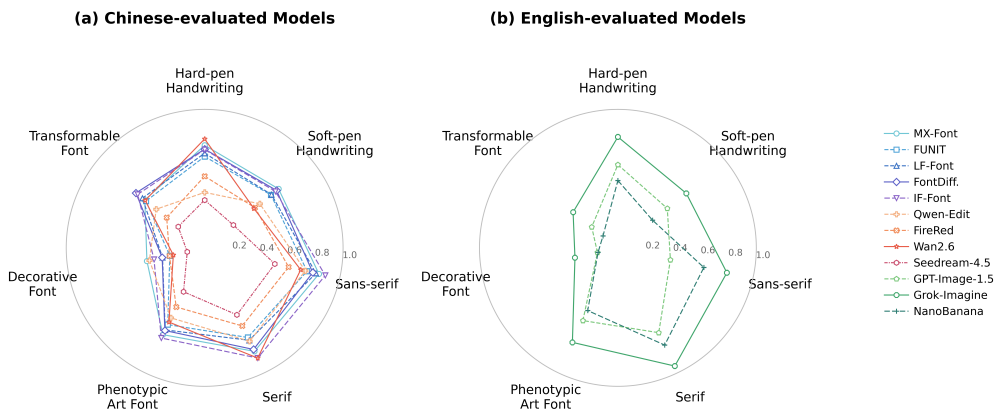


Figure 7: Category-wise normalized aggregate performance for (a) Chinese-evaluated and (b) English-evaluated models. Metrics are direction-aligned and normalized independently within each evaluation track, first averaged within the three FontBench evaluation dimensions and then equally aggregated across dimensions. Higher values indicate better relative performance within the corresponding track. The aggregate score is used only for visualization and should not be interpreted as a direct comparison between the Chinese and English tracks. The figure reports the seven font categories for which category-wise measurements are available.

art, and transformable fonts. Their performance nevertheless decreases for several more challenging styles, with decorative and soft-pen handwriting fonts showing lower scores for multiple methods.

The general-purpose Chinese image-editing models exhibit stronger category dependence. Qwen-Image-Edit-2511 and FireRed-Image-Edit-1.1 achieve competitive scores for several regular or moderately stylized categories, but their performance decreases on decorative and soft-pen styles. Wan2.6-Image shows comparatively strong results on hard-pen handwriting and serif fonts, while its performance is less uniform across the remaining categories. Doubao-Seedream-4.5 exhibits the largest overall degradation among the evaluated Chinese models, particularly on decorative and soft-pen handwriting fonts. These results suggest that high general image-generation capability does not necessarily translate into uniform robustness across typographic styles.

The specialized methods also exhibit different category-specific strengths rather than a single model consistently dominating every font type. For example, their relative performance changes across printed, handwritten, and artistic categories. This variation indicates that a single model-level average can obscure important differences in style-specific robustness and motivates the category-aware evaluation provided by FontBench.

A similar pattern is observed for the English-evaluated models in Fig. 7(b). Grok-Imagine-Image achieves the highest aggregate score across most of the displayed categories, with particularly strong results on serif and hard-pen handwriting styles. GPT-Image-1.5 shows intermediate performance, whereas Nano Banana 2 is more category-dependent. Across the three models, serif, hard-pen handwriting, and phenotypic art fonts are generally handled more reliably, while decorative and transformable styles yield lower aggregate scores.

Importantly, the values in the two panels should not be compared directly. The Chinese and English tracks contain different scripts, model sets, and evaluation distributions, and their normalization statistics are computed independently. The figure therefore characterizes *relative category-wise behavior within each track*, rather than absolute cross-language model superiority.

Overall, the category-wise results reveal performance heterogeneity that is not visible from aggregate model-level results alone. Regular or moderately stylized fonts are generally easier to reproduce, whereas decorative, highly deformed, and expressive handwritten styles expose larger differences between methods. This observation supports the use of fine-grained font-category analysis as a complementary component of FontBench.

F Additional Quanlitative Results

F.1 Decorative Fonts

Figure 8 presents a representative example from the decorative-font category. The target font (Font22) is characterized by thin strokes, irregular handwritten geometry, curved terminals, and local ornamental details. Successful generation therefore requires both correct glyph construction and preservation of these fine-grained stylistic characteristics.

The specialized font-generation models exhibit substantially different structural behaviors. MX-Font preserves the identity of most target characters, although some ornamental details are simplified. In contrast, FUNIT and LF-Font exhibit clear structural failures on this case. FUNIT distorts the internal stroke organization of several characters, with particularly severe component deformation in "妹", "模", and "整". LF-Font similarly preserves relatively simple characters but introduces component entanglement and malformed substructures as character complexity increases. Its outputs for "模" and "整", in particular, deviate substantially from the correct glyph topology. These errors correspond to failures of character construction rather than merely imperfect style transfer.

FontDiffuser and IF-Font show a different pattern. Their outputs remain largely recognizable and preserve the major glyph topology, but the generated shapes are noticeably more regular than the target. Curved terminals, irregular local geometry, and other decorative details are partially suppressed. Thus, these methods tend to preserve structural correctness while regularizing the target style.

General-purpose image-editing models reveal another set of failure modes. Qwen-Image-Edit-2511 introduces pronounced curls, loops, and extended strokes in several characters. In some cases, these additional structures modify the underlying glyph geometry rather than simply reproducing the target ornamentation. Moreover, its outputs vary substantially across characters: while some glyphs are heavily stylized, others such as "模" and "整" become thick and regularized. This indicates both over-stylization and intra-font style inconsistency.

FireRed-Image-Edit-1.1 preserves the target character identity more consistently, but still exhibits substantial variation in stroke appearance and visible artifacts in individual outputs. Seedream provides an especially clear example of the distinction between readability and style preservation: all five characters are highly recognizable and structurally regular, yet their thick, conventional printed appearance differs strongly from the thin and ornamental target font. Wan2.6-image retains more of the target's lightweight and irregular stroke characteristics while maintaining recognizable glyph structures, although some fine decorative details are still simplified.

The English results exhibit a similar separation between structural correctness and stylistic fidelity. GPT-Image-1.5 preserves the identities of the target letters and their lightweight appearance, but tends toward a more calligraphic form. Grok-Imagine-Image produces highly recognizable letters with considerably thicker and rounder strokes, weakening the lightweight decorative characteristics of the target. Nano Banana 2 introduces more explicit flourishes and curved terminals, capturing the high-level decorative nature of the font, but the strength and geometry of these ornaments vary across letters.



Figure 8: Representative generation results for a decorative font (Font22). The examples reveal distinct failure modes in structural fidelity and style preservation. In particular, some methods fail to preserve the target glyph structure, while others generate highly readable characters but regularize or exaggerate the target decorative style.

Overall, this case exposes several qualitatively different failure modes: FUNIT and LF-Font exhibit direct glyph-structure errors; FontDiffuser and IF-Font preserve character identity but regularize decorative details; Qwen-Image-Edit-2511 can over-stylize glyphs and alter their structure; and Seedream achieves high readability while largely losing the target font identity. These observations demonstrate that character correctness, style preservation, and perceptual quality are not interchangeable properties, providing qualitative support for the multi-dimensional evaluation adopted in FontBench.

F.2 TRANSFORMABLE FONTS

Figure 9 presents a representative example from the transformable-font category. Unlike regular printed or handwritten fonts, the source font deliberately modifies conventional glyph geometry through non-uniform stroke deformation, asymmetric component shapes, altered spatial proportions, and locally transformed character skeletons. The main challenge is therefore to preserve the identity and topology of the target character while reproducing these intentional geometric transformations.

The specialized font-generation models exhibit substantially different behaviors under this structurally deformed style. MX-Font preserves the identity of most target characters and retains part of the irregular stroke geometry of the source font. However, several transformed local structures are simplified, indicating that the model tends to preserve the canonical character skeleton more strongly than the precise geometric deformations of the source style.

FUNIT exhibits clear structural failures on several complex characters. Although relatively simple characters such as "中" remain recognizable, the generated "妹", "模", and "整" contain malformed components, incorrect stroke connections, and distorted spatial arrangements. These errors cannot be explained merely as imperfect style transfer: the underlying character topology itself is no longer faithfully preserved. This suggests that FUNIT has difficulty distinguishing intentional style-induced deformation from the structural constraints required to maintain a valid Chinese glyph.

LF-Font shows a similar pattern. Simple characters retain recognizable global structures, whereas more complex glyphs suffer from severe local component corruption. In "妹", "模", and "整", several strokes are merged, displaced, or reorganized in ways that depart from the correct character structure. This indicates that component-level style modeling becomes unstable when the source font strongly transforms the geometry of individual radicals and their spatial relationships.

FontDiffuser demonstrates better structural stability in this example. Most generated glyphs remain recognizable and preserve the major topological organization of the target characters. Nevertheless, the outputs often reproduce the source deformation only approximately: extreme local shape changes are reduced, and several transformed components become more regular. Its behavior therefore reflects a trade-off between maintaining valid glyph structure and preserving the full degree of geometric transformation present in the source font.



Figure 9: Representative generation results for a transformable font. The source glyphs exhibit deliberate geometric deformation, irregular component proportions, and transformed local structures. The examples reveal distinct failure modes, including incorrect glyph topology, excessive structural regularization, over-transformation, and visual artifacts.

IF-Font exhibits an even stronger tendency toward structural regularization. The generated glyphs are clean and largely preserve the canonical organization of the target characters, but many of the source-specific transformations are suppressed. Asymmetric components, irregular stroke expansions, and altered proportions are replaced by more standardized character forms. This

constitutes a typical case of *structural preservation with transformation underfitting*: the characters are correct, but the intended geometric deformation is only partially transferred.

The general-purpose Chinese image-editing models reveal different failure modes. Qwen-Image-Edit-2511 produces highly expressive glyphs and retains some of the source font's irregular geometry. However, several outputs contain excessive stroke expansion, noisy contours, or additional local structures. In these cases, the model does not simply reproduce the source transformation but further exaggerates it, leading to *over-transformation* and occasional structural deviation.

FireRed-Image-Edit-1.1 preserves the intended character identity more consistently, but local geometric instability remains visible. Some generated glyphs contain blurred contours, gray residual regions, or inconsistent stroke widths, and complex characters still exhibit local structural deviations. Thus, even when the global character identity is maintained, the transformed geometry is not reproduced uniformly.

Doubao-Seedream-4.5 provides a particularly informative contrast. Its generated characters are highly legible and structurally regular, but the distinctive transformations of the source font are largely removed. The outputs become thick, conventional, and print-like, with more standard component proportions and stroke arrangements. This case shows that high readability and structural correctness do not necessarily imply successful reproduction of a transformable font style.

Wan2.6-image provides a more balanced result in this example. Most target characters remain recognizable while retaining part of the source font's non-uniform stroke geometry and irregular component shapes. Nevertheless, some of the stronger local transformations are still simplified, suggesting that precise preservation of style-specific geometric deformation remains challenging even when the overall visual appearance is relatively close to the source.

The English-evaluated models exhibit the same tension between valid character topology and transformed style geometry. GPT-Image-1.5 correctly renders all target letters and preserves relatively thin and irregular strokes, but several letters become more conventional and calligraphic than the source. The model therefore maintains character identity while partially regularizing the target transformation.

Grok-Imagine-Image also produces highly recognizable letters, but its outputs are substantially thicker and more geometrically regular. The distinctive transformed silhouettes of the source letters are weakened, indicating that the model prioritizes canonical letter construction over precise reproduction of the source deformation.

Nano Banana 2 behaves differently. It attempts to preserve stronger local transformations, but in several cases the deformation becomes structurally unstable. The generated "B" contains a large dark region together with additional contour traces, while "M" exhibits overlapping and malformed stroke structures. The generated "y" is also heavily distorted. These examples show that reproducing aggressive geometric transformation without sufficient structural constraints can directly damage letter topology and readability.

Overall, this transformable-font case exposes a central difficulty in font generation: models must distinguish between *intentional geometric deformation*, which defines the target style, and *unintended structural corruption*, which changes the character itself. FUNIT and LF-Font exhibit direct glyph-topology failures; FontDiffuser and IF-Font preserve valid character structures but underreproduce the intended transformations; Qwen-Image-Edit-2511 occasionally exaggerates the deformation beyond the source style; Doubao-Seedream-4.5 produces highly readable but strongly regularized glyphs; and Nano Banana 2 demonstrates that aggressive transformation can lead to severe structural artifacts.

These qualitatively different behaviors further motivate the multi-dimensional evaluation adopted in FontBench. Structural Fidelity is required to distinguish valid style-induced deformation from malformed character topology, Style Consistency measures whether the intended geometric transformation is actually preserved, and Visual Quality captures artifacts, contour noise, and unintended local distortions. A single readability or perceptual-similarity metric cannot adequately characterize these complementary failure modes.

F.3 PHENOTYPIC ART FONTS

Figure 10 presents a representative example from the phenotypic art-font category. In contrast to decorative fonts, whose style is often characterized by explicit ornamental elements, and transformable fonts, which modify glyph geometry more aggressively, this category is primarily distinguished by its overall visual phenotype. The source font exhibits heavy stroke mass, irregular contour thickness, rounded local shapes, and a strong sense of visual weight. Successful generation therefore requires not only preserving character identity, but also reproducing the global silhouette and stroke-mass distribution that define the font's appearance.

For the specialized Chinese font-generation models, MX-Font preserves the overall character identity and retains part of the source font's heavy visual appearance, but the generated contours are more regular and less compact than those of the source. FUNIT produces recognizable glyphs for several characters, yet its stroke mass is noticeably reduced and the overall appearance becomes thinner and less expressive. LF-Font shows a larger departure from the source phenotype: several complex characters exhibit distorted local components, and the generated strokes no longer form the dense, coherent silhouettes observed in the target font.



Figure 10: Representative generation results for a phenotypic art font. The source style is characterized by heavy stroke mass, irregular contours, and a distinctive global visual silhouette. The comparison highlights differences in how models reproduce appearance-level characteristics beyond character correctness.

FontDiffuser preserves the heavy and irregular appearance more clearly than the thinner specialized baselines, although its local contours remain rough and some component shapes are simplified. IF-Font, in contrast, generates clean and structurally regular glyphs, but substantially reduces the source font's visual mass and irregularity. The resulting characters remain readable but appear more conventional, indicating that the model captures character structure more reliably than appearance-level expressiveness.

The general-purpose Chinese image-editing models show a different tendency. Qwen-Image-Edit-2511 produces glyphs with strong visual weight and therefore preserves the overall bold appearance of the source more effectively than methods that regularize the style. However, the generated contours are often noisy and uneven, with visible texture-like artifacts around stroke boundaries. FireRed-Image-Edit-1.1 maintains recognizable characters but shows inconsistent contour rendering across samples; some glyphs contain outline-like or gray residual regions that weaken the solid appearance of the source font.

Doubao-Seedream-4.5 generates highly legible and strongly weighted glyphs, but the source-specific irregularity is largely replaced by a cleaner and more standardized bold typeface. Its outputs therefore preserve the coarse visual weight of the target while losing part of the font's distinctive phenotypic identity. Wan2.6-image retains more irregular stroke shapes and non-uniform contour thickness, although its outputs are generally lighter than the source and do not fully reproduce the same dense visual mass.

The English results reveal a similar separation between readability and appearance-level style preservation. GPT-Image-1.5 produces correct and clean letterforms, but the strokes are thinner and more conventional than those of the source. Grok-Imagine-Image retains a heavier and more rounded silhouette, preserving more of the source font's visual mass, although the letter contours remain more regular. Nano Banana 2 generates highly readable letters but regularizes them substantially toward standard bold forms, reducing the irregular and hand-shaped appearance that characterizes the source font.

Overall, this case demonstrates that phenotypic art-font generation places particular emphasis on *global appearance preservation*. Several models can produce structurally correct and highly read-

able glyphs while still failing to reproduce the source font's stroke mass, contour irregularity, and silhouette. Conversely, models that preserve stronger visual expressiveness may introduce contour noise or unstable local appearance. These results show that character correctness alone cannot capture the quality of phenotypic style transfer, and further motivate the separate evaluation of Structural Fidelity, Style Consistency, and Visual Quality in FontBench.

F.4 HARD-PEN HANDWRITING

Figure 11 shows a representative example from the hard-pen handwriting category. Compared with printed fonts, hard-pen handwriting places less emphasis on strict geometric regularity and instead requires models to capture writing-specific characteristics, including thin and slightly irregular strokes, natural handwritten rhythm, subtle shape variation, and consistent stroke continuity. Unlike highly stylized artistic fonts, however, this category still preserves relatively clear character structures. The main challenge is therefore not severe deformation, but whether a model can maintain readability while reproducing the light, fluent, and personalized appearance of handwritten glyphs.

For the specialized Chinese font-generation models, MX-Font reproduces the overall handwritten tendency reasonably well. Its outputs remain legible and retain a relatively light stroke style, although some characters appear slightly simplified and less fluent than the source. FUNIT also preserves the thin handwritten appearance to some extent, but its generated glyphs show local instability, especially in more complex characters, where stroke connections and component shapes become less natural. LF-Font exhibits larger deviations: although the global style remains handwriting-like, several characters contain visibly distorted local structures, and the final glyphs appear less controlled and less coherent.

FontDiffuser provides comparatively stable results in this case. It preserves both readability and the overall handwritten impression, particularly in the slender stroke shapes and loose spatial arrangement. However, some outputs are slightly rough in contour quality and do not fully reproduce the same writing rhythm as the source. IF-Font generates clean and recognizable glyphs, but its outputs become more regularized than the source handwriting. In particular, the character "妹" shows a clear structural deviation, indicating that the model may sacrifice local character accuracy when trying to maintain a simplified handwritten style.



Figure 11: Representative generation results for a hard-pen handwriting font. The source style is characterized by thin strokes, slight geometric irregularity, and a natural handwritten rhythm. The examples illustrate differences in local structural accuracy, stroke-weight preservation, handwriting naturalness, and visual artifacts across specialized font-generation and general-purpose image-editing models.

The general-purpose Chinese image-editing models behave quite differently. Qwen-Image-Edit-2511 generates bold and visually striking glyphs, but this comes at the cost of style mismatch: the source hard-pen handwriting is light and thin, whereas the generated characters become heavy, brush-like, and over-emphasized. FireRed-Image-Edit-1.1 preserves the approximate layout of the characters, yet its outputs contain obvious outline artifacts and hollow stroke patterns, which weaken both the handwritten realism and the visual completeness of the glyphs. Doubao-Seedream-4.5 produces highly readable characters, but its outputs are strongly regularized toward a standard bold printed appearance, substantially reducing the source font's handwritten identity. Wan2.6-image offers a more balanced result: it keeps the glyphs thin and readable, while preserving more of the light, handwritten style than the other general-purpose baselines, although the stroke shapes are still cleaner and more uniform than those of the source.

The English results show a similar pattern. Since alphabetic characters are structurally simpler, all three English image-editing models generate highly readable outputs. GPT-Image-1.5 produces neat and accurate letters, but the results are visually cleaner and more standardized than the source. Grok-Imagine-Image better preserves stroke thickness and handwritten impression, though the letters remain somewhat regularized. Nano Banana 2 retains the thin overall appearance, but some letters appear overly smooth or faint, reducing the sense of natural pen-written variation.

Overall, this example shows that hard-pen handwriting mainly tests a model's ability to preserve *writing-style naturalness* rather than only structural correctness. Many models can generate readable glyphs, yet fail to reproduce the light stroke quality, subtle irregularity, and personal writing rhythm that define the source style. In particular, general-purpose models often over-regularize the outputs into heavier or more standard printed forms, whereas specialized models better maintain the handwritten character of the source but may still struggle with local structural precision. This case further illustrates why handwritten fonts should be evaluated not only by readability, but also by style consistency and fine-grained structural fidelity.

F.5 SOFT-PEN HANDWRITING

Figure 12 presents a representative example from the soft-pen handwriting category. Compared with hard-pen handwriting, soft-pen handwriting is more challenging because its style is defined not only by readable character structure, but also by brush-like stroke modulation, heavy ink texture, rounded terminals, and clear variation between thick and thin strokes. Therefore, successful generation requires a model to preserve both character identity and the dynamic visual behavior of brush writing.



Figure 12: Representative generation results for a soft-pen handwriting font. This category is characterized by brush-like stroke modulation, heavy ink style, rounded terminals, and strong variation in stroke thickness. The examples highlight differences across models in readability, brush-style preservation, stroke continuity, and artifact control.

Among the specialized font-generation models, MX-Font captures the overall brush-like appearance relatively well. Its outputs remain readable and retain the bold handwritten tendency of the source, although some local details are still slightly simplified and the stroke transitions are less natural than those in the source. FUNIT also preserves the general visual weight of the font, but its outputs appear coarser and less stable, with noticeable irregularities in local shapes and contour quality. LF-Font further amplifies this problem: although the generated glyphs roughly follow the target content, several characters show obvious structural distortion and over-thickened local components, which weakens both readability and stylistic coherence.

FontDiffuser produces comparatively strong results in this example. It maintains good readability while preserving much of the source font's heavy brush texture and thick stroke appearance. However, some generated characters still look slightly rough, and the brush modulation is not always as smooth as in the reference. IF-Font gives the cleanest and most stable outputs among the specialized baselines. Its characters are highly readable and structurally well aligned, but the style is somewhat regularized: compared with the source, the generated glyphs appear cleaner and more controlled, with weaker expression of brush softness and ink-like variation.

The general-purpose Chinese image-editing models again show a different behavior pattern. Qwen-Image-Edit-2511 reproduces the heavy visual weight and bold appearance of the source, but the generated glyphs contain rough stroke boundaries and local over-rendering, making the outputs visually aggressive rather than naturally brush-written. FireRed-Image-Edit-1.1 shows the least stable behavior in this case. Although it preserves some global shape cues, the outputs contain obvious artifacts and

local failures, including abnormal interior structures and noisy regions, indicating poor robustness for complex brush-style rendering. Doubao-Seedream-4.5 generates highly legible characters with strong black stroke weight, but the outputs are more like bold printed glyphs than natural soft-pen handwriting. The brush dynamics are weakened, especially in the more uniform stroke edges and more rigid appearance. Wan2.6-image offers a better compromise: it keeps the glyphs readable and visually bold while preserving more of the source style's handwritten character, although the brush texture is still cleaner and less expressive than in the reference.

For the English examples, all three models successfully preserve character identity, confirming that alphabetic generation is structurally easier. However, stylistic fidelity remains nontrivial. GPT-Image-1.5 produces neat and readable letters, but the outputs are thinner and more regular than the source. Grok-Imagine-Image better preserves the thick and heavy appearance of the reference, making it the closest to the source style in overall impression. Nano Banana 2 also captures the bold style reasonably well, but some letters exhibit slight local inconsistency and residual background patches, indicating weaker output cleanliness.

Overall, this case shows that soft-pen handwriting is particularly effective for testing whether a model can reproduce *brush dynamics* rather than merely thick strokes. Many models are able to generate readable glyphs, but fewer can accurately preserve the continuous stroke modulation, ink-like texture, and soft handwritten rhythm that define this category. In particular, general-purpose models tend to convert the source style into a heavier and more standardized bold form, while specialized models are generally better at maintaining brush-style continuity, though they may still suffer from local deformation or over-smoothing. This case therefore complements the hard-pen example by showing that handwriting evaluation should consider not only readability and structure, but also the fidelity of stroke-level stylistic dynamics.

F.6 SERIF

Figure 13 shows a representative example from the serif category. Compared with artistic and handwriting fonts, serif fonts are structurally more regular, but they impose stricter requirements on fine-grained detail reproduction. In this category, successful generation depends not only on preserving the overall character structure, but also on accurately rendering thin strokes, serif terminals, stroke contrast, and balanced spatial layout. As a result, serif fonts are particularly useful for evaluating the precision and stability of font generation models.



Figure 13: Representative generation results for a serif font.

Among the specialized font-generation models, overall performance is relatively strong. MX-Font preserves the general structure and the thin, elongated appearance of the source font, although minor local artifacts can still be observed, especially around some stroke junctions. FUNIT also produces highly readable glyphs and preserves the global serif style reasonably well, but some local contours appear slightly simplified, indicating limited sensitivity to subtle stroke-ending details. LF-Font maintains the thin printed style in general, yet several characters show small local deviations in stroke proportion and shape, suggesting that its content preservation is less stable when fine structural precision is required.

FontDiffuser and IF-Font provide the most faithful results among the specialized baselines in this example. Both models preserve the thin-stroke appearance, clean structure, and regular proportions of the serif source font. Their outputs are visually close to the target style, with better control over stroke width and terminal details than the other specialized methods. This suggests that when the source font is structurally regular, specialized models are generally effective at capturing the core typographic properties of serif fonts.

The general-purpose Chinese image-editing models show a different behavior. Qwen-Image-Edit-2511 preserves character readability well, but the generated glyphs are much bolder than the source and lose the delicate serif details. Its outputs are therefore structurally correct but stylistically over-regularized. FireRed-Image-Edit-1.1 better preserves the thin overall appearance, yet it introduces obvious local artifacts, especially hollow or outlined rendering effects in some characters, which substantially reduce typographic realism. Doubao-Seedream-4.5 generates highly legible outputs, but its results again become much heavier and more uniform than the source, making them closer to standard bold printed glyphs than true serif forms. Wan2.6-image produces comparatively better style fidelity within the general-purpose Chinese models: the glyphs remain clean and readable, and the thin printed appearance is better preserved, although the outputs are still more regularized and less detailed than those of the best specialized models.

The English examples further confirm this trend. GPT-Image-1.5 generates clean and highly readable serif letters, preserving most of the global style. Grok-Imagine-Image also performs well, though its letters appear slightly heavier and more stylized in some cases. Nano Banana 2 shows less stable behavior: while some letters are generated correctly, others exhibit obvious local artifacts or mismatched rendering, such as hollow interiors, background patches, or exaggerated stroke treatment. This indicates that even for structurally simple alphabetic glyphs, preserving the subtle serif style remains nontrivial.

Overall, this case demonstrates that serif fonts provide an important test of *fine-detail fidelity*. Unlike decorative or handwriting fonts, their difficulty does not mainly come from large structural deformation, but from the need to reproduce subtle typographic cues with high precision. Most models can preserve readability in this category, but fewer can faithfully capture the delicate stroke contrast and serif terminals that define the source style. This explains why serif fonts remain an essential category for diagnosing whether a model truly supports precise font-style transfer rather than merely generating plausible readable glyphs.

F.7 Sans-serif Fonts

Figure 14 presents a representative case from the sans-serif category. Compared with handwritten and artistic fonts, sans-serif fonts exhibit lower stylistic complexity and more regular glyph construction. Their visual identity is primarily determined by uniform stroke weight, clean boundaries, balanced component proportions, and consistent geometric organization. Consequently, this category provides a useful test of whether a model can reproduce precise typographic structure without relying on strong decorative or handwritten cues.

The specialized Chinese font-generation models perform relatively stably on this regular style. MX-Font preserves the overall glyph structure and bold sans-serif appearance for most characters, although small isolated artifacts are visible around several outputs. FUNIT also maintains recognizable and geometrically regular glyphs, with only moderate variation in stroke thickness and local proportions. Compared with the more difficult artistic categories, its structural behavior is noticeably more stable in this example.

LF-Font preserves the main character topology but exhibits slightly less consistent local geometry, particularly in more complex glyphs, where some components become overly dense or unevenly weighted. FontDiffuser produces clean and recognizable outputs with relatively uniform strokes, while some glyphs appear marginally lighter or smoother than the source. IF-Font similarly generates highly regular characters and preserves the simple sans-serif structure well. Its outputs are visually clean, although the source-specific stroke weight and local proportions are somewhat standardized.

The behavior of the general-purpose Chinese image-editing models is also considerably more stable than in the handwritten or artistic cases. Qwen-Image-Edit-2511 preserves character identity across the displayed examples and reproduces the bold sans-serif appearance, but its strokes are generally heavier than those of the source. The results therefore remain stylistically plausible while showing a systematic shift in visual weight.

FireRed-Image-Edit-1.1 exhibits a more distinctive failure mode. Although the target characters remain recognizable, several outputs contain outline-like rendering, hollow interior structures, and residual gray or colored regions. These effects are especially

visible in characters such as “中” and “模”, where the solid, uniform stroke structure of the source is replaced by contour-like rendering. Thus, the primary limitation in this case is not character identity but output appearance and artifact control. Doubao-Seedream-4.5 produces highly readable and geometrically regular glyphs with strong stroke weight. Its outputs are close to a conventional bold sans-serif typeface, but the stroke mass is somewhat heavier and more standardized than that of the source. This again illustrates that correct character structure can coexist with measurable style deviation even for a relatively simple printed-font category.



Figure 14: Representative generation results for a sans-serif font.

Wan2.6-image generates clean and recognizable glyphs while retaining a more moderate stroke weight than Qwen-Image-Edit-2511 and Doubao-Seedream-4.5. The overall geometric organization is stable, and the outputs remain visually consistent across the displayed characters. Nevertheless, minor differences in stroke width and component proportion remain relative to the source.

The English results exhibit an analogous pattern. Because the source letters have simple sans-serif geometry, all three evaluated models preserve letter identity successfully. GPT-Image-1.5 produces clean and well-formed glyphs but generally uses thinner strokes than the source, resulting in a noticeable weight mismatch. Grok-Imagine-Image retains a heavier and more compact appearance that is closer to the source’s bold sans-serif character, although some letterforms remain slightly regularized. Nano Banana 2 also preserves the sans-serif structure and high readability, but stroke thickness and local proportions vary across letters.

Overall, the sans-serif case exhibits substantially fewer catastrophic structural failures than the handwritten, decorative, and transformable examples. The remaining differences are primarily reflected in *stroke-weight calibration*, *geometric consistency*, and *artifact suppression*. This observation indicates that regular printed fonts are comparatively easier for both specialized and general-purpose models, while also showing that high readability alone does not guarantee exact typographic reproduction. Even in this relatively simple category, models can generate correct characters while deviating in stroke width, component proportion, or rendering quality, supporting the need for complementary structural, stylistic, and perceptual evaluation.