UC-FONT: UNIFIED CONTENT REPRESENTATION FOR FEW-SHOT FONT GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-shot font generation (FFG) aims to synthesize stylistically consistent characters from limited reference glyphs. The font content sources are restricted to either standard-font glyph images or component sequences, such as radical sequences and stroke sequences, failing to support image and text modality inputs within a unified paradigm. In this paper, we propose UC-Font, a unified framework for FFG with versatile input compatibility. It contains a Unified-Content Representer (UCR) to unify character content representation through cross-modal alignment at local and global levels, enabling flexible input of both glyph images and structural component sequences. For style aggregation, conventional cross-attention fails to adequately suppress irrelevant features, causing stylistic inconsistencies between generated fonts and target fonts even for identical character components. To tackle this issue, we present a Temperature-Adjusted Style Aggregator (TASA), which leverages component similarity-conditioned attention to strengthen the style consistency of shared character components. UC-Font could be built upon diffusion-based and flow matching-based frameworks, where FFG is formulated as a conditional generation task with fused unified contents and aggregated style features serving as conditions. To enhance the content and style feature interaction through joint attention modeling for FFG, a Dual-Projection Fusion (DPF) module is adopted. Extensive experiments show that UC-Font outperforms state-of-the-art FFG methods.

1 INTRODUCTION

Few-shot font generation (FFG) aims to generate a complete font set with a consistent style from only a small number of reference glyphs. By leveraging modern learning-based techniques, FFG alleviates the time-consuming and labor-intensive process of traditional font design, particularly for complex scripts such as Chinese, Japanese, and Korean. Its efficiency and scalability make it well-suited for practical applications, including digital media, advertising, and gaming.

The paradigm of content-style disentanglement (Zhang et al., 2018b) addresses the limited generalization ability of image-to-image translation-based methods (Liu et al., 2019; Tian, 2017). These approaches typically require first rendering all target characters into standard font (i.e., source font) images and then disentangling content representations, which limits input flexibility. Additionally, achieving perfect disentanglement of content and style (Wang et al., 2023) is challenging. The style-agnostic character components (Fig. 1(a)), such as strokes (He et al., 2024) and radicals (Xue et al., 2021; Chen et al., 2024), are widely used to model glyph structures and fine-grained details (Kong et al., 2022; Park et al., 2024). While component representations supply semantic content priors, they lack explicit spatial and geometric information, which degrades performance and generalization on unseen characters.

To tackle the above limitations, we propose a novel FFG method termed UC-Font. At its core is the Unified-Content Representer (UCR) module, which supports unified content extraction from both image and text modalities. The text modality adopts style-agnostic Ideographic Description Sequences (IDS) with two granularities: radical-level (IDS-R) and stroke-level (IDS-S), collectively referred to as component sequences.

UCR employs three dedicated encoders to extract modality-specific content features from glyph images, IDS-R, and IDS-S. A set of learnable query tokens aggregates these heterogeneous features

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

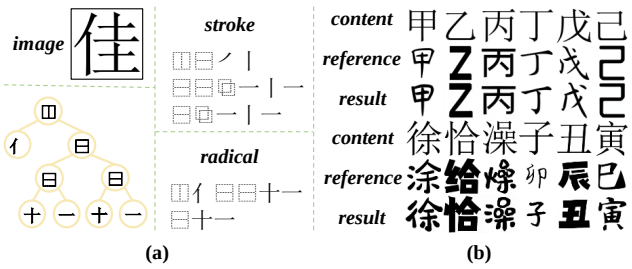

Figure 1: Existing problems for FFG. (a) Different character content representations. (b) Style inconsistency for sharing components.

into a single unified content representation via cross-attention. During training, a multi-modal alignment strategy enforces local and global consistency among the three modality-specific representations. Since all modalities encode identical character content and IDS features are style-invariant, this alignment suppresses style residuals in image-derived content features. Meanwhile, IDS-based representations gain precise spatial and geometric priors by aligning with glyph images. Such cross-modal interaction compensates for the inherent deficiencies of each single modality while exploiting their complementary advantages. Consequently, UCR enables flexible multi-modal input and robust unified content modeling.

Component-wise style representations (Pan et al., 2023; Yao et al., 2024) enable precise, flexible stylistic modeling by independently capturing the attributes of each glyph component. Most methods aggregate component styles via cross-attention with content as queries and reference style features as keys and values. Nevertheless, the softmax normalization in vanilla dot-product attention distributes non-zero weights to numerous irrelevant keys, resulting in redundant style aggregation. It introduces undesirable stylistic variations even for shared components in target characters (Fig. 1(b)).

Perceptually, generated glyph regions corresponding to shared components between target and reference characters should strictly inherit reference styles, while unshared components allow reasonable stylistic flexibility under global style consistency. Therefore, we propose the Temperature-Adjusted Style Aggregator (TASA), which adaptively adjusts attention softmax temperature based on component-level content similarity between target and reference glyphs. Higher component similarity lowers the temperature to produce sharper attention weight distributions, forcing the model to aggregate styles dominantly from the most matched reference components and eliminate interference from irrelevant style features.

To this end, a Dual-Projection Fusion (DPF) module is proposed to better fuse the unified content features and the aggregated style features through dual-branch projection. The fused features serve as conditions for Diffusion Model (DM)-based or Flow-matching (FM)-based generation networks. We extensively evaluate UC-Font on LDM (Rombach et al., 2022), Flux (Labs et al., 2025) and PixelDiT (Yu et al., 2026). Experimental results demonstrate that it can generate more stable and consistent font styles than other state-of-the-art (SOTA) methods.

We summarize our main contributions as follows: (1) We present UC-Font, a novel FFG model equipped with a UCR module that supports flexible input modalities for character content representation. (2) Motivated by human perceptual mechanisms for font design, we propose a novel TASA module that dynamically calibrates attention weights to mitigate stylistic redundancy. (3) Extensive experiments demonstrate the effectiveness of UC-Font for both seen and unseen font generation.

2 Related Works

Few-shot font generation. Early few-shot font generation (FFG) methods adopt CNN-(Krizhevsky et al., 2012) and GAN-based (Goodfellow et al., 2014) image-to-image translation paradigms. Representative approaches, including zi2zi (Tian, 2017) and DC-Font (Jiang et al., 2017), realize font stylization via image translation and feature reconstruction but suffer from weak generalization to unseen font. Subsequent studies disentangle global content and style embeddings for feature recombination. Methods such as EMD (Zhang et al., 2018b), DG-Font (Xie et al., 2021),

and SGD-Font (Li et al., 2025a) improve feature decoupling yet fail to capture fine-grained local style details. To remedy this, recent research shifts to component-level modeling based on strokes and radicals (Li & Lian, 2024; Xue et al., 2025). A series of advanced methods, including MX-Font (Park et al., 2021), FS-Font (Tang et al., 2022), VQ-Font (Yao et al., 2024), DeepCalliFont (Liu & Lian, 2024), IF-Font (Chen et al., 2024), FT-Font (Hong et al., 2025) and DML-Font (Sami et al., 2026), incorporate expert decomposition, cross-modal alignment, spatial correspondence, structural modeling, and diffusion-based generation. Despite these improvements, existing methods still struggle to maintain precise content fidelity for complex characters, frequently producing missing strokes and structural distortion artifacts.

Generation Model. GANs (Goodfellow et al., 2014) produce realistic images through adversarial learning, and conditional GANs (Mirza & Osindero, 2014) further enable generation control, achieving competitive performance in FFG (Wang et al., 2023; Tang et al., 2022; Wang et al., 2025). VAEs represent latent spaces as probability distributions; VQ-VAE (Van Den Oord et al., 2017) leverages discrete codebooks, and VQ-Font (Pan et al., 2023) adopts this quantization scheme to boost local style transfer for FFG. DMs generate images via iterative denoising with state-of-the-art synthesis capacity (Fu et al., 2024; Kim et al., 2024; Li & Zhu, 2025). Diff-font (He et al., 2024) takes strokes and style as generation conditions, while FontDiffuser (Yang et al., 2024) exploits multi-scale features for intricate glyphs. DMs surpass GANs and VAEs in training stability and output diversity, yet suffer from slow inference speed. FM delivers strong generative performance with markedly faster inference (Lipman et al., 2023; Chen & Lipman, 2024; Yao et al., 2025) and many approaches (Li et al., 2025b) recently adopt FM for the FFG task. PixelDiT (Yu et al., 2026) is a VAE-free dual-branch diffusion Transformer directly operating in pixel-space for high-fidelity glyph generation. However, they only accept image prompts as input, limiting multi-modal content control.

3 METHOD

The overall architecture of UC-Font is illustrated in Fig. 2. A Unified-Content Representer obtains a unified content representation F_c from the content input x_c, regardless of input type. Style features F_s, extracted from reference images I_s, are aggregated by the TASA module into content-aware style features F_{cs}. F_c and F_{cs} are further fused via DPF module to be generation conditions y for DM/FM networks. The full model is trained to generate images matching the content of x_c while conforming to the style of I_s.

3.1 PRELIMINARIES

Based on any DM/FM generation network (Ho et al., 2020; Song et al., 2021; Chen & Lipman, 2024), our aim is to train an FFG model with a guiding signal y, defined as the combination of the content features F_c and the style features F_{cs}, to better control the generated font images.

Given a paired training sample $(x_1, y) \sim q(x_1, y)$, where $x_1 \in \mathbb{R}^d$ is a font image from the training set, the forward diffusion process of the DM-based model (Rombach et al., 2022) progressively perturbs x_1 with Gaussian noise: $z_t = \sqrt{\bar{\alpha}_t} z_1 + \sqrt{1-\bar{\alpha}_t}\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$, where $\bar{\alpha}_t$ is determined by a predefined noise schedule and t denotes the discrete diffusion timestep. The FFG model predicts the noise ϵ from the latent representation z_t of x_1. During inference, it progressively denoises an initial Gaussian noise sample to obtain the generated font representation.

For the FM-based (Chen & Lipman, 2024) model, a time-dependent velocity field is learned to transport samples from a Gaussian noise distribution to the conditional target font distribution. Given an initial noise sample $x_0 \sim \mathcal{N}(0, I)$, we adopt a linear interpolation path (Liu et al., 2023): $x_t = (1-t)x_0 + tx_1$, $t \in [0, 1]$. The corresponding target velocity is $\frac{\mathrm{d}x_t}{\mathrm{d}t} = x_1 - x_0$. A neural network $u_t^\theta(x_t \mid y)$ is trained to predict this velocity. During inference, the generation starts from $x_0 \sim \mathcal{N}(0, I)$ and proceeds by solving the ordinary differential equation (ODE) (Chen et al., 2018).

3.2 UNIFIED-CONTENT REPRESENTER

The proposed UCR module unifies character content representations and allows users to select the content input modality among character images, IDS-R, and IDS-S.

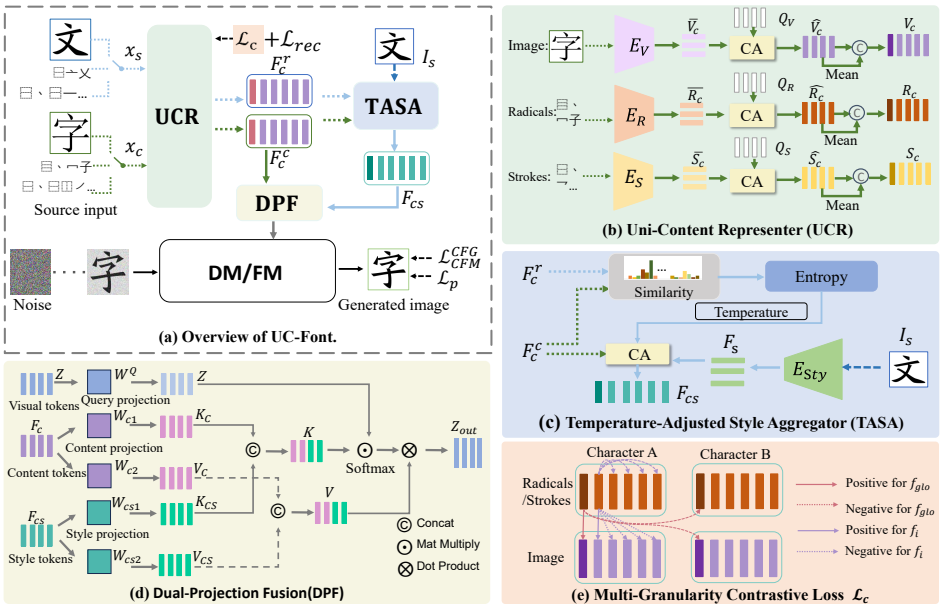

Figure 2: (a) Overview of our proposed UC-Font. It is a DM/FM-based network composed of a UCR, TASA and DCCA modules for FFG. (b) UCR extracts content features F_c from content input x_c in any modality. (c) TASA module obtains the font style representations F_{cs} from reference image I_s. (d) DPF projects the content and style features dually in a unified cross-attention. (e) Multi-Granularity Contrastive Loss $\mathcal{L}_c$ optimizes content representations at both the global and local levels via contrastive learning.

For images as content input, we extract the visual representation of the character using a CNN-based encoder, denoted as $\bar{V}_c = \{v_1, v_2, \ldots, v_I\}$, where I is the number of image tokens. Radical and stroke sequences are two description sequences of characters at different semantic levels. Given these two sequences, we separately employ a Transformer-based encoder (Devlin et al., 2019) to obtain the radical-level and stroke-level representations, denoted as $\bar{R}_c = \{r_1, r_2, \ldots, r_J\}$ and $\bar{S}_c = \{s_1, s_2, \ldots, s_K\}$, respectively. J and K represent the lengths of the radical and stroke sequences, respectively. The radical encoder and the stroke encoder share the same architecture, each consisting of L_2 Transformer layers.

To unify character content representations across different modalities, we adopt CLIP-guided optimization (Yu et al., 2023) to align IDS features with character image features. However, CLIP aligns only global features, which is insufficient for capturing fine-grained local features of characters. Additionally, the token lengths of the same character across different modalities differ, and they also vary across characters for IDS-R and IDS-S. Thus, it is difficult to directly align the token-level features $\bar{V}_c$, $\bar{R}_c$, and $\bar{S}_c$. Therefore, we design three types of learnable queries, i.e., Q_V, Q_R, and Q_S, and use cross-attention to extract local representations $\hat{V}_c$, $\hat{R}_c$, and $\hat{S}_c$ from the three content representations, respectively. This process can be formulated as:

$$\hat{F}_c = \mathrm{CA}(Q_F, \bar{F}_c, \bar{F}_c), F \in \{V, R, S\}, \tag{1}$$

where $\mathrm{CA}(\cdot)$ denotes a cross-attention layer followed by a feed-forward network. $\hat{F}_c = \{f_1, f_2, \ldots, f_N\}, f_n \in \mathbb{R}^d$. Note that each modality has the same number of queries. Additionally, the local representation tokens in $\hat{F}_c$ are averaged to obtain the global content representation f_{glo}. Finally, we form the unified local and global content representation $F_c = \{f_1, f_2, \ldots, f_N, f_{glo}\}$. We leverage contrastive learning to align features from different modalities and encourage UCR to output unified representations.

3.3 Temperature-Adjusted Style Aggregator

Our TASA module aims to ensure that if the target and reference characters share components, the model extracts local style precisely from the corresponding reference component. Relevant style

cues are identified and aggregated via hard attention. Otherwise, soft attention is adopted to infer style by aggregating information from multiple similar local structures.

To achieve this goal, the TASA module takes as input the content representations of the reference and target characters, $F_c^r = \{f_1^r, f_2^r, \ldots, f_N^r, f_{glo}^r\}$ and $F_c^c = \{f_1^c, f_2^c, \ldots, f_N^c, f_{glo}^c\}$, as well as the reference style features $F_s = \{f_1^s, f_2^s, \ldots, f_M^s\}$ extracted by a style encoder. Then it aggregates local styles into content-aware style representations F_{cs} via temperature-adjusted cross-attention.

Formally, we first compute the cosine similarity $S(i,j)$ between the content tokens of the target character F_c^c and those of the reference character F_c^r:

$$S(i,j) = \frac{f_i^c \cdot f_j^r}{\| f_i^c \| \| f_j^r \|}. \tag{2}$$

We then apply a Softmax over j for each i to obtain a similarity distribution $\bar{S}(i,j) \in \mathbb{R}^{(N+1)\times(N+1)}$. To measure the concentration of this distribution, we compute its entropy and use it as an adaptive temperature factor:

$$\tau_i = -\sum_{j=1}^{N+1} \bar{S}(i,j) \log \bar{S}(i,j). \tag{3}$$

A lower τ_i indicates a higher local similarity between the target and reference characters. Therefore, style should be extracted from a small subset of tokens in F_s; otherwise, style should be aggregated in a more distributed manner. We incorporate the temperature into the attention score to obtain:

$$A(i,j) = \frac{\exp(\lambda\, q_i \cdot k_j / \tau_i)}{\sum_{m=1}^{M} \exp(\lambda\, q_i \cdot k_m / \tau_i)}, \tag{4}$$

where q_i is the i-th token of the query matrix $Q_c \in \mathbb{R}^{(N+1)\times d^s}$ and k_j is the j-th token of the key matrix $K_s \in \mathbb{R}^{M\times d^s}$. Here, Q_c is obtained by projecting F_c^c, and K_s and V_s are obtained by projecting F_s with linear layers. We set $\lambda = \alpha/\sqrt{d^s}$, where α is a positive learnable parameter.

With TASA, we relate content similarity to the attention sharpness for style extraction, and obtain the aggregated style features: $F_{cs} = AV_s$. Finally, the aggregated style F_{cs} is concatenated with the content representation F_c along the sequence dimension, and used as the generation condition.

3.4 Dual-Projection Fusion

For the FFG task, content conditions and style conditions have different semantic functions and feature distributions. If shared projection layers are used after direct concatenation, feature interference between modalities may occur; while injecting the two types of conditions completely independently makes it difficult to model fine-grained correlations between content and style. Therefore, we use a DPF module, which introduces two sets of trainable linear projection layers $W_{1,2}^{c/cs}$ to handle content features F_c and style features F_{cs} separately instead of directly concatenating them. W^Q is the linear projection of query feature Z. We denote Z_{out} as the output latent feature. While preserving the distinctions between content and style features, this design maintains cross-modal interactions between the two conditions and image states. It helps alleviate feature interference caused by shared mapping and enhances the conditional expressiveness of character structure and font style. Then, the process of DPF can be formulated as follows:

$$K = \text{Concat}\big(F_c W_{c1},\, F_{cs} W_{cs1}\big), V = \text{Concat}\big(F_c W_{c2},\; F_{cs} W_{cs2}\big) \tag{5}$$

$$Z_{\text{out}} = \text{Softmax}\left(\frac{(ZW^Q)K^\top}{\sqrt{d}}\right) V \tag{6}$$

3.5 Training Objective

We adopt supervised multi-granularity contrastive loss (Khosla et al., 2020) $\mathcal{L}_c$ to realize cross-modal alignment and feature consistency for UCR. It is composed of two parts: global contrastive loss $\mathcal{L}_g$ and local contrastive loss $\mathcal{L}_l$.

For global feature learning, positive pairs consist of features from the same character across modalities, while features of different characters form negative pairs. Given batch unified content features $\{F_c^i\}_{i=1}^B$, $F_c^i = \{f_1^i, \cdots, f_N^i, f_{glo}^i\}$ denotes modality-agnostic content features, and $ch(\cdot)$ returns the corresponding character label. The global contrastive loss is defined as:

$$\mathcal{L}_g = \frac{1}{B}\sum_{i=1}^{B}\frac{-1}{|P_g(i)|}\sum_{p\in P_g(i)}\log\frac{\exp(z_i\cdot z_p)}{\sum_{a\in A_g(i)}\exp(z_i\cdot z_a)}, \tag{7}$$

where $z_i = \frac{f_{glo}^i}{\|f_{glo}^i\|}$, $A_g(i) \equiv \{1\dots B\}\setminus\{i\}$, and $P_g(i) \equiv \{p \in A_g(i) \mid ch(f_{glo}^p) = ch(f_{glo}^i)\}$ is the set of indices of all positive samples.

For learning the local feature representation contrastively, we consider the token-wise relationship in $\hat{F}_c$. Two tokens are considered as positive pairs if their character categories and indices are identical, and negative pairs if they share the same character category but differ in index. To simplify the formulation, we flatten all local features within the batch and notate them as $\{f_k^l\}_{k=1}^{B\times N} \equiv \{f_1^1, \cdots, f_N^1, f_1^2, \cdots, f_N^B\}$. Analogous to eq. (7), the contrastive loss for local features is formulated as:

$$\mathcal{L}_l = \frac{1}{B\times N}\sum_{k=1}^{B\times N}\frac{-1}{|P_l(k)|}\sum_{p\in P_l(k)}\log\frac{exp(z_k\cdot z_p)}{\sum_{a\in A_l(k)}exp(z_k\cdot z_a)}, \tag{8}$$

where $z_k = \frac{f_k^l}{\|f_k^l\|}$, $A_l(k) \equiv \{a \in \{1,2,\dots,B\times N\} \mid a \neq k, ch(f_a^l) = ch(f_k^l)\}$, and $P_l(k) \equiv \{p \in A_l(k) \mid ID(f_p^l) = ID(f_k^l)\}$ is the set of indices of all positive samples. $ID(f_k^l)$ denotes the index of f_k^l in F_c, which is used to distinguish different vectors within the same set of content features. The multi-granularity contrastive loss is computed as the sum of two terms: $\mathcal{L}_c = \mathcal{L}_g + \mathcal{L}_l$.

Additionally, we introduce a reconstruction loss as an auxiliary constraint. Specifically, we employ an image decoder $\mathrm{D}_{\mathrm{img}}$ and a text decoder $\mathrm{D}_{\mathrm{IDS}}$ to reconstruct the corresponding image and IDS (i.e., IDS-R, IDS-S) from the unified representation. $\mathrm{D}_{\mathrm{img}}$ follows the decoder design proposed in (Yu et al., 2024), while $\mathrm{D}_{\mathrm{IDS}}$ adopts Transformer layers for autoregressive generation. The reconstruction loss functions are formulated as:

$$\mathcal{L}_{img} = \|D_{img}(F_c) - I_c\|_2^2, \quad \mathcal{L}_{IDS} = -\sum_{t=1}^{J}\log p_\theta(r_t \mid r_{<t}, F_c), \tag{9}$$

$$\mathcal{L}_{rec} = \alpha_{img}\mathcal{L}_{img} + \alpha_{IDS}\mathcal{L}_{IDS}, \tag{10}$$

where I_c and $r_{1:J}$ denote the image and IDS labels corresponding to F_c, p_θ represents the probability distribution predicted by $\mathrm{D}_{\mathrm{IDS}}$ and $\alpha_{img}, \alpha_{IDS}$ are the weighting coefficients for the loss terms.

To further improve the font generation quality, we adopt the perceptual loss (Zhang et al., 2018a) for style transfer learning. It is worth noting that the perceptual loss is computed in the pixel space. Therefore, the model's predicted output, such as noise and velocity field, must first be converted into the corresponding target image before loss computation. The detailed derivation of this transformation process is provided in the Appendix A. The corresponding loss function is defined as follows:

$$\mathcal{L}_p = \sum_{l=1}^{L}\left\|\varphi_l(\tilde{x}_1) - \varphi_l(x_1)\right\|, \tag{11}$$

where φ_l is the feature of l_{th} layer from VGG (Simonyan & Zisserman, 2015) and L is the total number of the chosen layers.

In addition, the generative objective $\mathcal{L}_{gen}$ for training the base DM or FM-based models is used according to the adopted generation paradigm: $\mathcal{L}_{\mathrm{CFM}}^{\mathrm{CFG}}$ for FM-based model (Zheng et al., 2023; Ho & Salimans, 2021) and $\mathcal{L}_{\mathrm{Diff}}^{\mathrm{CFG}}$ for DM-based model (Ho et al., 2020). More details can be found in the Appendix B.

The overall loss function is defined as the weighted sum of all the aforementioned loss terms:

$$\mathcal{L}_{all} = \mathcal{L}_{gen} + \mathcal{L}_c + \mathcal{L}_{rec} + \alpha_p\mathcal{L}_p. \tag{12}$$

Table 1: Quantitative comparison with state-of-the-art methods on SFUC and UFUC datasets. The best results are highlighted in bold, and the second-best results are underlined.

Methods	SFUC					UFUC				
	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓
DG-Font	33.8962	0.3682	0.3793	0.2226	0.4284	44.5148	0.3729	0.3588	0.2334	0.4402
FS-Font	58.9956	0.3089	0.5052	0.1501	0.3600	90.2807	0.3916	0.2872	0.2352	0.4409
VQ-Font	18.6300	0.2445	0.5296	0.1406	0.3440	28.1769	0.3469	0.4022	0.2096	0.4283
CF-Font	31.0571	0.3515	0.4112	0.2089	0.4136	39.9225	0.3483	0.4124	0.2030	0.4052
FontDiffuser	10.2242	0.3042	0.4652	0.1737	0.3636	20.6686	0.3377	0.4135	0.1935	<u>0.3897</u>
IF-font	7.3169	0.2351	0.5038	0.1566	0.3656	18.4763	<u>0.2943</u>	<u>0.4256</u>	0.1893	0.4087
FSTDiff	10.1215	0.2937	0.4806	0.1656	0.3516	18.4361	0.3104	**0.4408**	**0.1759**	**0.3662**
zi2zi-JiT	80.9716	0.3500	0.3800	0.2220	0.4381	87.9352	0.3444	0.3670	0.2236	0.4393
Ours Flux(Image)	<u>3.9966</u>	<u>0.1936</u>	<u>0.5544</u>	<u>0.1300</u>	<u>0.3215</u>	**15.7533**	**0.2922**	0.4248	<u>0.1857</u>	0.4001
Ours Flux(IDS-R)	4.1269	0.2030	0.5421	0.1366	0.3310	17.0694	0.3047	0.4148	0.1950	0.4088
Ours Flux(IDS-S)	**3.9801**	**0.1921**	**0.5563**	**0.1290**	**0.3199**	<u>16.4667</u>	0.3107	0.4063	0.1997	0.4131

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. To comprehensively evaluate the performance of our proposed method, we constructed a dataset comprising 351 Chinese fonts (a diverse collection of standard print and artistic font styles) and 3100 commonly used Chinese characters. Following the standard setting, 330 fonts and 3000 characters are used for training, while 20 fonts and 100 characters are used as unseen fonts and characters, respectively. Following the commonly adopted settings (Wang et al., 2023; Yang et al., 2024), we evaluate the generation performance of 100 unseen characters on both SFUC (Seen Font Unseen Character) and UFUC (Unseen Font Unseen Character).

Metrics. We follow the settings of most FFG methods (Wang et al., 2023; Yang et al., 2024), using FID (Heusel et al., 2017), LPIPS (Zhang et al., 2018a), SSIM (Wang et al., 2004), L1, and RMSE as evaluation metrics.

Implementation Details. Our primary model adopts Flux (Labs et al., 2025) with flow matching (Chen & Lipman, 2024). During training, we randomly drop the condition with a probability of $p_{\text{uc}} = 0.1$, and use classifier-free guidance with a scale of $w = 4.5$ during inference. The Flux-based FM model is optimized with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$) using a OneCycleLR scheduler, with a base learning rate of 4.5×10^{-6}, a batch size of 40, and 50 training epochs on a single NVIDIA A40 GPU. Since UC-Font could be applied to different generation models, we further build it with LDM (Rombach et al., 2022) and PixelDiT (Yu et al., 2026). Their detailed training and inference configurations and results are provided in the Appendix C and Appendix D.

4.2 COMPARISON WITH STATE-OF-THE-ART METHODS

We compare our method with eight state-of-the-art (SOTA) methods: DG-Font (Xie et al., 2021), FS-Font (Tang et al., 2022), VQ-Font (Yao et al., 2024), CF-Font (Wang et al., 2023), FontDiffuser (Yang et al., 2024), IF-Font (Chen et al., 2024), FSTDiff (Li & Zhu, 2025) and zi2zi-JiT (Tian, 2026). Among them, DG-Font is an image-to-image translation-based method. FontDiffuser and IF-Font primarily rely on global style modeling, using glyph images and IDS as content inputs, respectively. FS-Font, VQ-Font, and CF-Font consider local feature representations. FSTDiff and zi2zi-JiT capture global and local style details for FFG. For fair comparison, we select SimSun, a widely used standard typeface, as the source font, and all methods are trained based on their official codes and settings. We fix all outputs as 128×128 RGB images and normalize the pixel values to the range $[0, 1]$. Since our method can accept image, IDS-R, or IDS-S as content input, we report three results by individually using each of them.

Quantitative comparison. The quantitative results of one-shot font generation are presented in Table 1. Compared with other methods on SFUC, our model with IDS-S as input achieves the best performance on all metrics. Especially, our model with any content input modality achieves

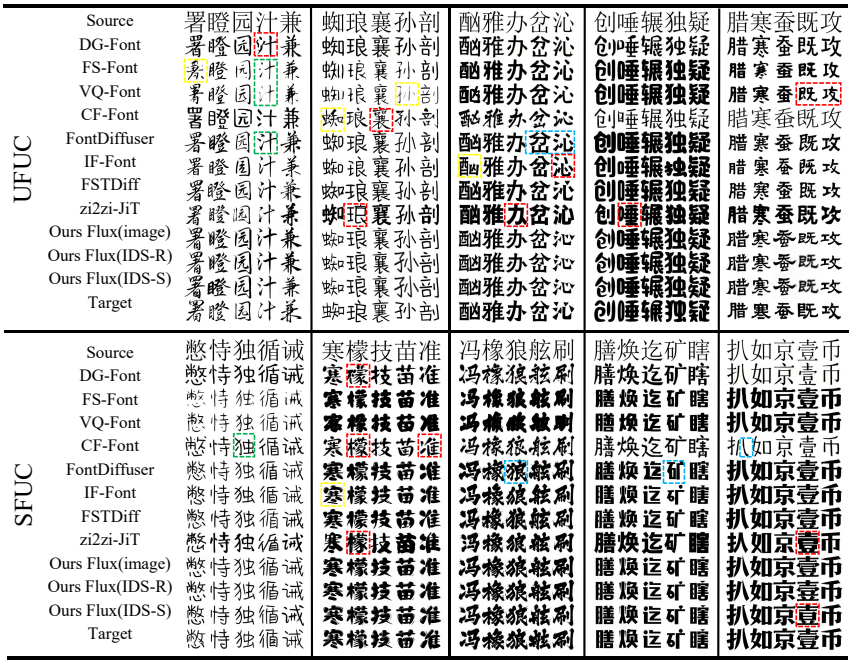

Figure 3: Qualitative comparison on SFUC and UFUC. Colored dashed boxes highlight typical failure cases: green indicates errors on very thin-stroke styles, blue marks missing fine-grained details, red denotes disordered radicals/strokes, and yellow denotes missing radicals/strokes.

significant improvement compared with other methods on FID, which indicates that UC-Font can transfer the styles to unseen characters better. For UFUC, our model also outperforms other methods on the FID metric, illustrating that generated fonts could better match the target font styles and are visually closer to human perception. When using an image as content input, our method gets the best FID and LPIPS scores. In the L1 metric, it achieves the second-best performance and nearly comparable results to the best ones. This result demonstrates the strong generalization of our model for new fonts.

Qualitative comparison. The visualizations of one-shot font generation results on SFUC and UFUC are presented in Fig.3. DG-Font and CF-Font struggle with style transfer under large style gaps between reference and content. FS-Font and VQ-Font struggle with very thin-stroke styles (see green dashed boxes), indicating difficulty in capturing fine-grained stroke-level style cues. FontDiffuser performs better than other methods in global visual perception, but it has a higher probability of failing to generate fine-grained details (see blue dashed boxes) compared with our method. IF-Font adopts IDS as content input, which may generate disordered (see red dashed boxes) and missing (see yellow dashed boxes) radicals or strokes. FSTDiff and zi2zi-JiT still exhibit erroneous glyph structures or mismatched font styles. In addition, most existing methods fail to reproduce complex connection strokes. However, our proposed UC-Font successfully handles these cases and focuses more on the local font details, demonstrating that UC-Font consistently generates high-quality fonts with superior content preservation, style consistency, and structural correctness.

4.3 Ablation Studies

Loss ablation. Table 2 shows ablation results of the proposed training objectives. Built upon the baseline trained with only $\mathcal{L}_{CFM}^{CFG}$, sequentially adding perceptual, contrastive, and reconstruction losses achieves persistent improvements. Comparisons of (a) vs (b) and (c) vs (d) reveal that $\mathcal{L}_p$ substantially lifts LPIPS, SSIM, L1 and RMSE to strengthen perceptual fidelity and overall consistency. Comparisons of (a) vs (c) and (b) vs (d) illustrate that $\mathcal{L}_c$ enhances structural accuracy by improving all metrics. The reconstruction loss $\mathcal{L}_{rec}$ further boosts all metrics when comparing (d) and (e). The consistent improvements confirm the robustness and effectiveness of training losses.

Table 2: Ablation study on different training losses.

	Loss				IDS-R					Image				
	$\mathcal{L}_{CFM}^{CFG}$	$\mathcal{L}_p$	$\mathcal{L}_c$	$\mathcal{L}_{rec}$	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓
(a)	✓				22.1062	0.3437	0.3694	0.2274	0.4387	21.8057	0.3472	0.3779	0.2149	0.4323
(b)	✓	✓			18.6456	0.3167	0.3993	0.2058	0.4208	18.4768	0.3129	0.4076	0.1979	0.4164
(c)	✓		✓		18.7029	0.3285	0.4021	0.2061	0.4215	18.9088	0.3107	0.4156	0.1978	0.4139
(d)	✓	✓	✓		19.1279	0.3228	0.4047	0.2019	0.4159	19.08941	0.3047	0.4143	0.1955	0.4093
(e)	✓	✓	✓	✓	**17.0694**	**0.3047**	**0.4148**	**0.1950**	**0.4088**	**15.7533**	**0.2922**	**0.4248**	**0.1857**	**0.4001**

Module ablation. Table 3 reports the quantitative ablation results of TASA and DPF in UC-Font. To validate the effectiveness of the TASA module, we replace it with vanilla cross-attention, where no adjustment is applied to the attention distribution. Comparisons of (a) vs (c) reveal that adopting TASA leads to consistent improvements in the model across all metrics. The qualitative results of the TASA module are presented in the Appendix F. To verify the contribution of the DPF module, we simply concatenate F_c and F_{cs}, and apply shared projection weights. Comparisons of (a) vs (b) demonstrate the effectiveness of the DPF module in fusing content and style features. Moreover, jointly incorporating TASA and DPF yields further improvements across the evaluation metrics, demonstrating the benefits of their combined use in improving structural fidelity and stylistic consistency for high-quality font generation.

Table 3: Ablation study on TASA and DPF modules.

	Module		IDS-R					Image				
	TASA	DPF	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓
(a)			18.6118	0.3097	0.4041	0.2003	0.4134	18.5143	0.2976	0.4140	0.1956	0.4105
(b)		✓	17.9325	0.3088	0.4093	0.1985	0.4130	16.7428	0.2963	0.4176	0.1911	0.4064
(c)	✓		17.3637	0.3064	0.4112	0.1974	0.4127	15.9643	0.2952	0.4224	0.1885	0.4048
(d)	✓	✓	**17.0694**	**0.3047**	**0.4148**	**0.1950**	**0.4088**	**15.7533**	**0.2922**	**0.4248**	**0.1857**	**0.4001**

Input modality ablation. Without UCR, the encoders $\mathrm{E_V}$, $\mathrm{E_R}$, and $\mathrm{E_S}$ are trained separately to produce modality-specific content representations. Such variants are denoted as X-only models, $X \in \{Image, IDS-R, IDS-S\}$. As shown in Table 4, under this configuration, the generated results vary significantly across different content modalities, indicating that the DM/FM model struggles to adapt to heterogeneous modality features. We attribute this to the lack of constraints among the three types of content, causing their representations to lie in distinct feature spaces. When the UCR module is applied, the results across all three modalities become more consistent, demonstrating that UCR effectively aligns them within a unified representation space.

Table 4: Ablation study on content input modalities with and without UCR.

Content	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓
Image-only/UCR(Image)	17.4491/15.7533	0.3039/0.2922	0.4146/0.4248	0.1924/0.1857	0.4055/0.4001
IDS-R-only/UCR(IDS-R)	22.8750/17.0694	0.3977/0.3047	0.3211/0.4148	0.2582/0.1950	0.4803/0.4088
IDS-S-only/UCR(IDS-S)	22.6605/16.4667	0.3991/0.3107	0.3219/0.4063	0.2588/0.1997	0.4814/0.4131

5 CONCLUSION

This paper present a novel FFG model, i.e., UC-Font. By introducing the Unified-Content Representer (UCR), Temperature-Adjusted Style Aggregator (TASA), and Dual-Projection Fusion (DPF), our method achieves robust cross-modal content alignment and precise style aggregation. Extensive experiments demonstrate that UC-Font outperforms existing methods on seen and unseen fonts, delivering superior style consistency, structural fidelity, and generalization ability. **Limitation:** Our method may still generate characters with missing strokes (as shown in Fig.3 and the Appendix J). We attribute this partly to imperfect IDS decomposition that introduces noise, and to components varying in appearance across spatial locations, which can hinder IDS-based generation. In future work, we will investigate more robust IDS-aware representations and more effective cross-modality collaboration to further advance the FFG task.

AI USE STATEMENT

In this paper, we use AI tools only to aid in writing and polishing the text, including improving clarity, grammar, and readability. We did not rely on AI to create any scientific content, design experiments, process experimental data, or derive conclusions. All technical ideas, experiments, and results reported in this paper were completed by the authors.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. This work does not involve human subjects, animal experiments, or sensitive private data collection. All datasets adopted in our experiments contain no human portraits, personal identity information, or any sensitive private data. We strictly follow the ICLR Code of Ethics throughout our research, and our model is only developed for academic research.

REPRODUCIBILITY STATEMENT

We devote significant efforts to facilitating the reproducibility of all results in this paper. To facilitate reproducibility, we will release the complete source code, training scripts, model checkpoints, detailed configuration files, and datasets upon paper acceptance. We report full hyperparameter settings, training schedules, hardware environments, and all quantitative evaluation metrics without omission.

REFERENCES

Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, pp. 47922–47945, 2024.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pp. 6572—6583, 2018.

Xinping Chen, Xiao Ke, and Wenzhong Guo. IF-Font: ideographic description sequence-following font generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 14177–14199, 2024.

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

Bin Fu, Fanghua Yu, Anran Liu, Zixuan Wang, Jie Wen, Junjun He, and Yu Qiao. Generate like experts: multi-stage font generation by incorporating font transfer process into diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6892–6901, 2024.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672—-2680, 2014.

Haibin He, Xinyuan Chen, Chaoyue Wang, Juhua Liu, Bo Du, Dacheng Tao, and Yu Qiao. Diff-Font: diffusion model for robust one-shot font generation. *International Journal of Computer Vision*, 132:5372–5386, 2024.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6629—-6640, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Yang Hong, Yinfei Li, Xiaojun Qiao, and Junsong Zhang. Conditional font generation with content pre-train and style filter. *Computer Graphics Forum*, 44(1):e15270, 2025.

Yue Jiang, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. DCFont: An end-to-end deep chinese font generation system. In *SIGGRAPH Asia 2017 Technical Briefs*, pp. 1–4. ACM, 2017.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673, 2020.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. FIFO-Diffusion: generating infinite videos from text without training. In *Advances in Neural Information Processing Systems*, volume 37, pp. 89834–89868, 2024.

Yuxin Kong, Canjie Luo, Weihong Ma, Qiyuan Zhu, Shenggao Zhu, Nicholas Yuan, and Lianwen Jin. Look closer to supervise better: one-shot font generation via component-based discriminator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13472–13481, 2022.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. FLUX.1 Kontext: flow matching for in-context image generation and editing in latent space, 2025. URL `https://arxiv.org/abs/2506.15742`.

Hua Li and Zhouhui Lian. HFH-Font: few-shot chinese font synthesis with higher quality, faster speed, and higher resolution. *ACM Transactions on Graphics*, 43(6), 2024.

Shilin Li and Anna Zhu. FSTDiff: one-shot font generation via cross-font style transformation learning. In *International Conference on Document Analysis and Recognition*, pp. 167—182, 2025.

Zhenhua Li, Siyi Chen, and Dong Liang. SGD-font: style and glyph decoupling for one-shot font generation. *Knowledge-Based Systems*, pp. 114600, 2025a.

Zheyu Li, Honglie Wang, Yan-Ming Zhang, and Cheng-Lin Liu. FLUX-Font: stylized font generation based on fine-tuning pre-trained diffusion transformer. In *Pattern Recognition and Computer Vision*, pp. 14273–14282, 2025b.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

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. 10550–10559, 2019.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Yitian Liu and Zhouhui Lian. DeepCalliFont: few-shot chinese calligraphy font synthesis by integrating dual-modality generative models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3774–3782, 2024.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. URL `https://arxiv.org/abs/1411.1784`.

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.

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, 2021.

Song Park, Sanghyuk Chun, Junbum Cha, Bado Lee, and Hyunjung Shim. Few-shot font generation with weakly supervised localized representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(3):1479–1495, 2024.

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. 10674–10685, 2022.

Abdul Sami, Irfanullah Memon, Avinash Kumar, Jongsun Choi, and Jaeyoung Choi. DML-Font: multilingual font generation based on diffusion model. *KSII Transactions on Internet & Information Systems*, 20(5):2525, 2026.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Licheng Tang, Yiyang Cai, Jiaming Liu, Zhibin Hong, Mingming Gong, Minhu Fan, Junyu Han, Jingtuo Liu, Errui Ding, and Jingdong Wang. Few-shot font generation by learning fine-grained local styles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7895–7904, 2022.

Yuchen Tian. zi2zi: Master chinese calligraphy with conditional adversarial networks, 2017. URL `https://github.com/kaonashi-tyc/zi2zi`.

Yuchen Tian. zi2zi-JiT: font synthesis with pixel space diffusion transformers. GitHub repository, 2026. URL `https://github.com/kaonashi-tyc/zi2zi-jit`.

Aaron Van Den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pp. 6309—-6318, 2017.

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.

Xuanhong Wang, Cong Li, Zengguo Sun, and Luying Hui. RS-GAN: unsupervised running script font generation via disentangled representation learning and contextual transformer. *Pattern Analysis and Applications*, 28(2):97, 2025.

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.

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. 5126–5136, 2021.

Mobai Xue, Jun Du, Jianshu Zhang, Zi-Rui Wang, Bin Wang, and Bo Ren. Radical composition network for chinese character generation. In *International Conference on Document Analysis and Recognition*, pp. 252–267. Springer, 2021.

Mobai Xue, Jun Du, Zhenrong Zhang, Jiefeng Ma, Qikai Chang, Pengfei Hu, Jianshu Zhang, and Yu Hu. Skeleton and font generation network for zero-shot chinese character generation, 2025. URL `https://arxiv.org/abs/2501.08062`.

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.

Jixun Yao, Yuguang Yang, Yu Pan, Ziqian Ning, Jianhao Ye, Hongbin Zhou, and Lei Xie. StableVC: style controllable zero-shot voice conversion with conditional flow matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25669–25677, 2025.

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*, pp. 16407–16415, 2024.

Haiyang Yu, Xiaocong Wang, Bin Li, and Xiangyang Xue. Chinese text recognition with a pre-trained clip-like model through image-ids aligning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11943–11952, 2023.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 128940–128966, 2024.

Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. PixelDiT: pixel diffusion transformers for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018a.

Yexun Zhang, Ya Zhang, and Wenbin Cai. Separating style and content for generalized style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8447–8455, 2018b.

Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky T. Q. Chen. Guided flows for generative modeling and decision making, 2023. URL https://arxiv.org/abs/2311.13443.

A Latent-to-Pixel Space Reconstruction for Perceptual Supervision

Since DM-based and FM-based models perform generation in the latent space, the predicted outputs cannot be directly used for pixel-space perceptual supervision. Therefore, we first recover the clean latent representation z_t from the predicted noise or velocity field, and then decode it into the image space through the VAE decoder. For the DM-based model, the predicted noise component $\epsilon_\theta(z_t, t, y)$ is defined as described in 3.1. Thus, the latent representation can be estimated by:

$$\hat{z}_1 = \frac{z_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(z_t, t, y)}{\sqrt{\bar{\alpha}_t}}. \tag{13}$$

For the FM-based model, the velocity field $u_t^\theta(z_t|y)$ is optimized to approximate v_t, as defined in 3.1. According to the definition of the linear path, the latent representation z_t can be recovered as:

$$\hat{z}_1 = z_t + (1-t)u_t^\theta(z_t|y). \tag{14}$$

After obtaining the reconstructed latent representation $\hat{z}_1$, both generation paradigms share the same decoding process:

$$\tilde{x}_1 = D_{\mathrm{VAE}}(\hat{z}_1), \tag{15}$$

where $D_{\mathrm{VAE}}(\cdot)$ denotes the VAE decoder and $\hat{x}_1$ is the reconstructed font image in pixel space. Therefore, we can use pixel-space perceptual supervision L_p defined in Eq. equation 11.

B Generative Objective

For FM-based models, we train the velocity predictor with CFM (Zheng et al., 2023) objective and classifier-free guidance (Ho & Salimans, 2021):

$$\mathcal{L}_{\mathrm{CFM}}^{\mathrm{CFG}} = \mathbb{E}_{t,\xi,q(z_1,y),p(z_0)}\left[\left\|u_t^\theta(z_t \mid y_\xi) - (z_1 - z_0)\right\|^2\right], \tag{16}$$

where $t \sim \mathcal{U}[0,1]$, $z_0 \sim \mathcal{N}(0, I)$, and $\xi \sim \mathrm{Bernoulli}(p_{uc})$. The condition with classifier-free dropout is defined as

$$y_\xi = (1-\xi)y + \xi \cdot \emptyset \tag{17}$$

For the diffusion variant, we use LDM noise-prediction objective (Rombach et al., 2022) with the same classifier-free guidance (Ho & Salimans, 2021):

$$\mathcal{L}_{\mathrm{Diff}}^{\mathrm{CFG}} = \mathbb{E}_{z_1,\epsilon,t,\xi}\left[\|\epsilon_\theta(z_t, t \mid y_\xi) - \epsilon\|_2^2\right], \qquad \epsilon \sim \mathcal{N}(0, I). \tag{18}$$

C Additional Experimental Details

Flux-based DM Configuration. We adopt the same pretrained Flux checkpoint as the Flux-based FM model and reinitialize the final prediction layer to adapt the model from velocity prediction to noise prediction. The model is trained with the LDM (Rombach et al., 2022) noise-prediction objective using 1,000 diffusion timesteps and a linear noise schedule. The base learning rate is set to 3.90625×10^{-7}, with a batch size of 64 and 30 training epochs. We use the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, together with a OneCycleLR scheduler. During inference, we employ DDIM sampling with 50 denoising steps. Classifier-free guidance is applied with a condition dropout probability of $p_{\mathrm{uc}} = 0.1$ and a guidance scale of $w = 4.5$. The feature dimension is fixed to 256, and the generated images are 128×128 RGB images. The weights of the auxiliary training objectives are set to $\alpha_{\mathrm{IDS}} = 1$, $\alpha_{\mathrm{img}} = 0.05$, and $\alpha_{\mathrm{p}} = 0.01$. All experiments are conducted on a single NVIDIA A40 GPU.

PixelDiT-based FM Configuration. We use PixelDiT as the velocity predictor under the same flow matching framework as the Flux-based FM model. The model is trained with a base learning rate of 4.5×10^{-6}, a batch size of 40, and 50 training epochs. We use the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, together with a OneCycleLR scheduler. Classifier-free guidance is applied with a condition dropout probability of $p_{\mathrm{uc}} = 0.1$ and a guidance scale of $w = 4.5$. The feature dimension is fixed to 256, and the generated images are 128×128 RGB images. The weights of the auxiliary training objectives are set to $\alpha_{\mathrm{IDS}} = 1$, $\alpha_{\mathrm{img}} = 0.05$, and $\alpha_{\mathrm{p}} = 0.01$. All experiments are conducted on a single NVIDIA A40 GPU.

Table 5: Quantitative results between different models on the SFUC test set.

Methods	Source	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓
FLUX-based FM	image	4.0515	0.1881	0.5647	0.1257	0.3143
	IDS-R	4.2249	0.1977	0.5521	0.1322	0.3238
	IDS-S	4.0531	0.1882	0.5644	0.1258	0.3145
PixelDiT-based FM	image	4.2647	0.2141	0.5263	0.1412	0.3431
	IDS-R	4.2647	0.2141	0.5263	0.1412	0.3431
	IDS-S	4.2590	0.2142	0.5260	0.1415	0.3437
FLUX-based DM	image	3.9966	0.1936	0.5544	0.1300	0.3215
	IDS-R	4.1269	0.2030	0.5421	0.1366	0.3310
	IDS-S	3.9801	0.1921	0.5563	0.1290	0.3199

Table 6: Quantitative results between different models on the UFUC test set.

Methods	Source	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓
FLUX-based FM	image	15.7533	0.2922	0.4248	0.1857	0.4001
	IDS-R	17.0694	0.3047	0.4148	0.1950	0.4088
	IDS-S	16.4667	0.3107	0.4063	0.1997	0.4131
PixelDiT-based FM	image	16.8532	0.2985	0.4185	0.1893	0.4082
	IDS-R	17.0993	0.3102	0.4028	0.1991	0.4200
	IDS-S	16.9919	0.2999	0.4162	0.1905	0.4097
FLUX-based DM	image	16.7369	0.2996	0.4190	0.1918	0.4069
	IDS-R	16.9397	0.3070	0.4102	0.1977	0.4140
	IDS-S	16.8412	0.3006	0.4173	0.1928	0.4081

D Comprehensive Comparison of All Model Combinations

In the main text, we present the results of Flux-based FM. This section provides comprehensive quantitative metrics and qualitative visual comparisons for Flux-based DM and PixelDiT-based FM to fully evaluate all model combinations.

The quantitative results of one-shot font generation in Table 5 and Table 6 reveal consistent performance of our three models. The visualizations of one-shot font generation results are presented in Fig.4. Through quantitative and qualitative results on the UFUC test set, Flux-based FM with image as input achieves the best performance on all metrics. PixelDiT-based FM with image as input attains the second-best performance on L1 and LPIPS. Flux-based DM with image as input takes second place on SSIM, RMSE, and FID. On the SFUC test set, Flux-based DM with IDS-S as input attains the best FID score, whereas Flux-based FM with image as input outperforms all competitors on the remaining metrics. Flux-based FM with IDS-S as input achieves the second-best results on LPIPS, SSIM, L1, and RMSE. Flux-based DM with IDS-S as input ranks second in FID. Therefore, directly learning the transport velocity between the Gaussian source and the structured font latent distribution is effective for conditional glyph generation. In particular, the continuous transport formulation preserves content structure and style consistency more effectively. The diffusion variant nevertheless demonstrates competitive performance on FID, indicating that standard noise prediction remains a viable formulation. Its iterative denoising process provides a stable optimization target but requires a predefined noise schedule and repeated reverse updates. In contrast, FM directly learns a continuous vector field and permits generation using an ODE solver with a flexible number of function evaluations.

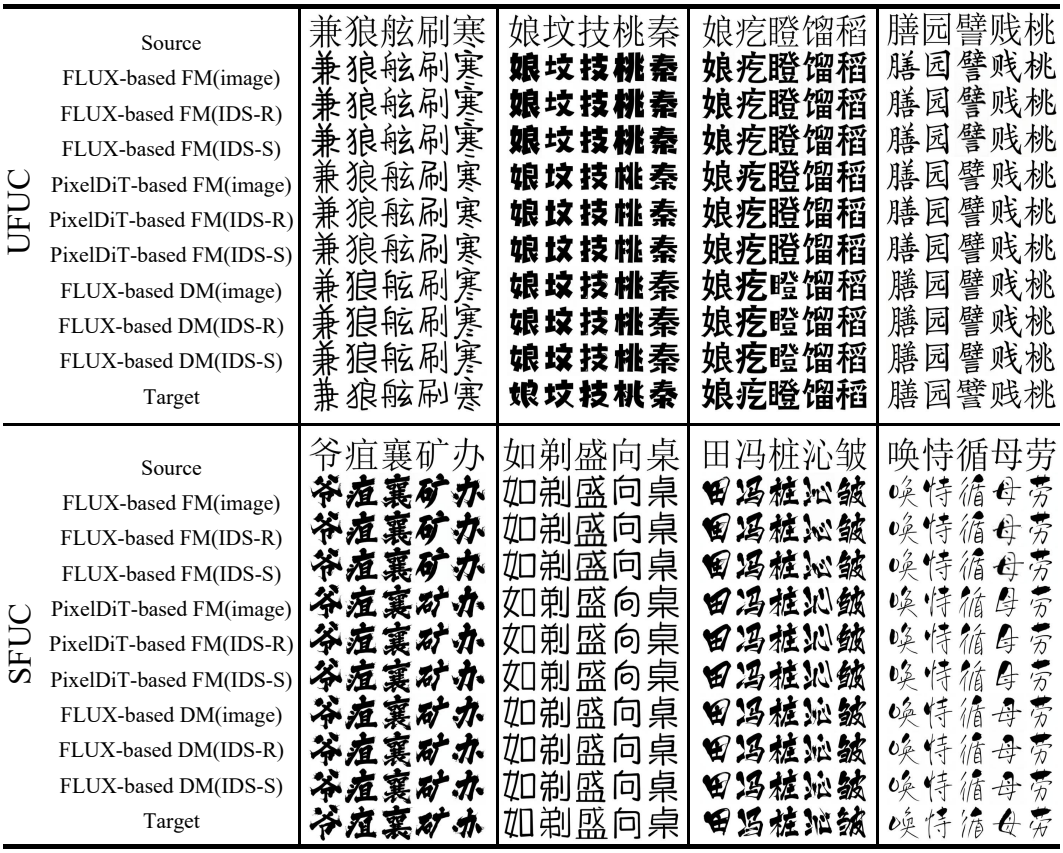

Figure 4: Qualitative comparison on SFUC and UFUC.

E Influence of Reference Numbers

The number of reference images is an important factor influencing the performance of the model. As the number of reference images increases, the model can capture richer style information. Our model is trained only under the 1-shot setting and directly generalizes to the k-shot setting without additional training. Table 7 presents the results when $k = 3$ and $k = 5$. The results show that as k increases, the LPIPS, SSIM, L1, and RMSE metrics improve, while FID slightly decreases. The improvement in LPIPS, SSIM, L1, and RMSE indicates better reconstruction accuracy for single-image generation, whereas the slight degradation in FID suggests that the generated results become more concentrated in specific styles or detail distributions, reducing diversity. Furthermore, as shown in Fig.5, when more reference images are provided, the generated results become more visually consistent with the ground truth (GT).

Table 7: Influence of Reference Numbers on UFUC.

k-shot	IDS-R					Image				
	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓
1	17.0694	0.3047	0.4148	0.1950	0.4088	15.7533	0.2922	0.4248	0.1857	0.4001
3	16.1962	0.2948	0.4204	0.1886	0.4040	16.0636	0.2874	0.4277	0.1832	0.3975
5	16.3461	0.2938	0.4205	0.1884	0.4038	16.2164	0.2864	0.4303	0.1822	0.3962

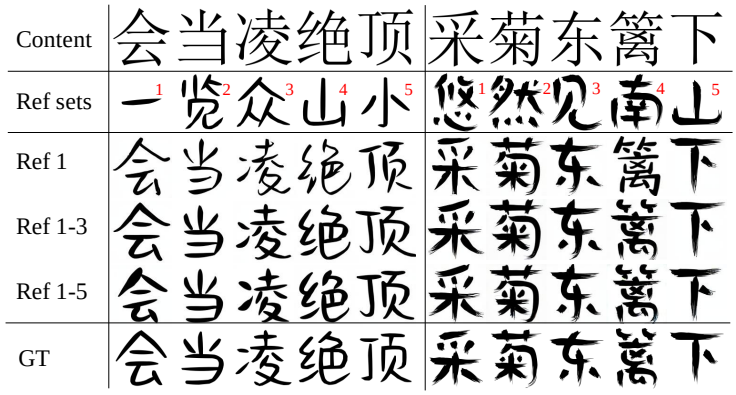

Figure 5: Few-shot generation results. The red numbers denote the IDs of the reference characters, and Ref 1-3 indicates that reference characters 1, 2, and 3 are used.

F Qualitative Analysis of TASA Module

For qualitative analysis, we visualize pairs of content and reference characters that share common components as illustrated in Fig.6. We observe that incorporating TASA allows the model to focus more accurately on these shared components, resulting in generated regions that better match the style of the reference image. Fig.6 uses a reference character that shares partial components with the target character. As a result, the effect is more evident in Fig.6.

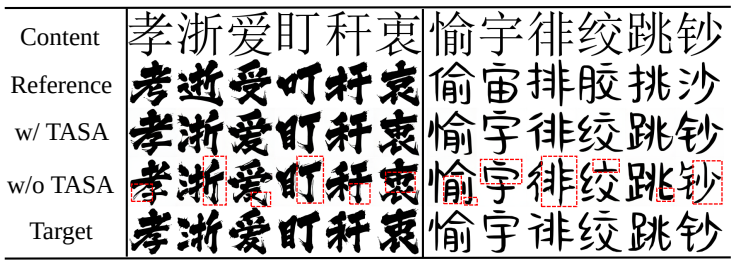

Figure 6: FFG results of our proposed model w/ and w/o TASA module. Red dashed boxes indicate style inconsistency, even though the corresponding parts are almost identical in the reference image and the target.

G Influence of Query Token Number in UCR

The number of queries mirrors the diversity of local script representations. To train the UCR module, we experiment with varying query counts, and the corresponding results are presented in Fig.7. As observed, the FID score dips slightly at 72 and 80 tokens but remains largely stable with further increases in token count. Notably, the lowest LPIPS value is achieved with 72 tokens, indicating optimal perceptual quality. Meanwhile, other metrics exhibit minimal fluctuations, suggesting that the pixel-level generation performance of our model is robust to token quantity, neither degrading nor improving drastically across the test range. Overall, we select 72 query tokens as the optimal setting for the UCR module.

H Training Strategy

Our model consists of two main components: the unified-content representer (UCR) module and the image generation module (TASA, DPF and Flow Matching). The UCR module is responsible for encoding content information, which is then used as a conditional input for the generation

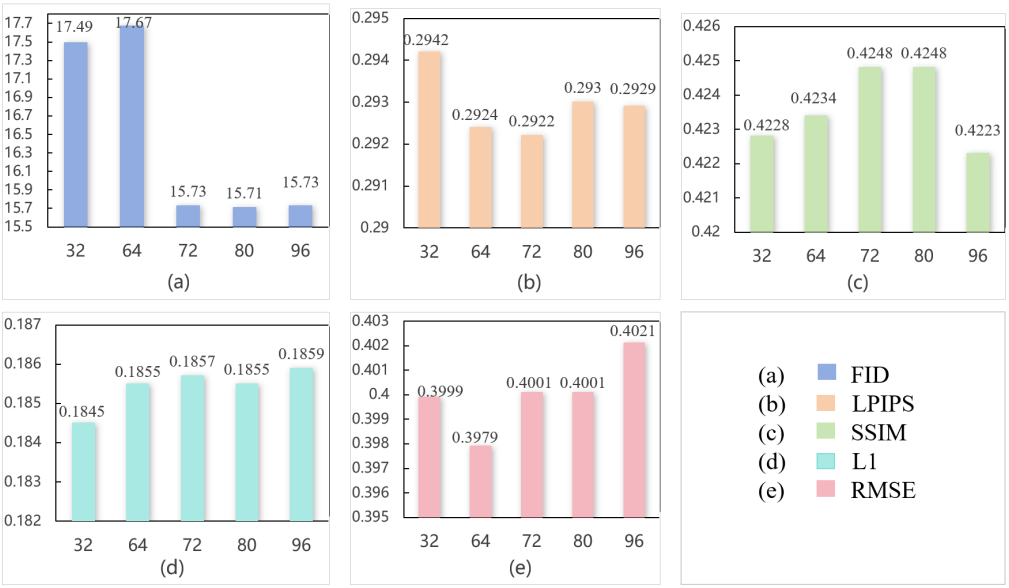

Figure 7: The influence of different query token numbers for FFG with image as source input.

module. Since UCR focuses solely on content representation, it can be independently trained using contrastive loss and reconstruction loss, followed by a second stage in which the generation module is trained. Alternatively, the entire model can be trained in an end-to-end manner. Table 8 reports the results on the UFUC dataset under both two-stage training and end-to-end training. The results show that end-to-end training consistently outperforms the two-stage approach across all evaluation metrics. This improvement may be attributed to the additional supervisory signals introduced by the flow-matching model, which helps refine the unified content representation during joint optimization.

Table 8: Influence of training strategy on UFUC: comparison between Two-Stage and End-to-End training.

Mode	IDS-R					Image				
	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓	FID ↓	LPIPS ↓	SSIM ↑	L1 ↓	RMSE ↓
Pre-train	18.0368	0.3047	0.4131	0.1946	0.4078	18.1039	0.2989	0.4184	0.1910	0.4034
End-to-end	17.0694	0.3047	0.4148	0.1950	0.4088	15.7533	0.2922	0.4248	0.1857	0.4001

I CROSS-LINGUAL FFG

To verify the generalization and cross-lingual FFG ability of the proposed method, we test our model on Japanese and Korean characters. Chinese characters differ substantially from Japanese and Korean scripts in certain structural components—for example, Chinese characters lack the circular elements frequently present in Hangul, so the direct generation performance tends to degrade in such cross-script settings. To address this issue, we fine-tune both the UCR module and the flow matching module to better accommodate these cross-lingual structural variations.

Specifically, for each linguistic script, we use a single font containing 300 characters to fine-tune the model, which aims to force the model to recognize the structures of different scripts. During testing, unseen character images from the corresponding script are used as content inputs, while Chinese fonts serve as reference images for cross-lingual font generation. This scenario is the same for testing on unseen content, but the content and references come from different domains. Some visualization results are shown in Fig.8. We can see that the generated cross-lingual characters successfully capture the stroke dynamics and terminal details characteristic of Chinese fonts.



Figure 8: Cross-language results on both Japanese and Korean.

It should be noted that our method uses only Japanese and Korean character images as content inputs and does not rely on IDS. This suggests that the proposed approach can be naturally extended to other scripts for which IDS is unavailable. In such cases, generation can be performed by simply using character images as the content input.

J More Visualization Results



Figure 9: Failure cases.

Fig.10 and Fig.11 present the comparative results of different methods across a broader range of font styles. Fig.9 collects several failure cases. As shown in Fig.9, IDS-based methods tend to produce more errors. We attribute this to the imperfect IDS decomposition rules, which introduce noise into the representation. Moreover, the same component can exhibit considerable visual differences when appearing at different spatial locations, further affecting the performance of IDS-based generation.

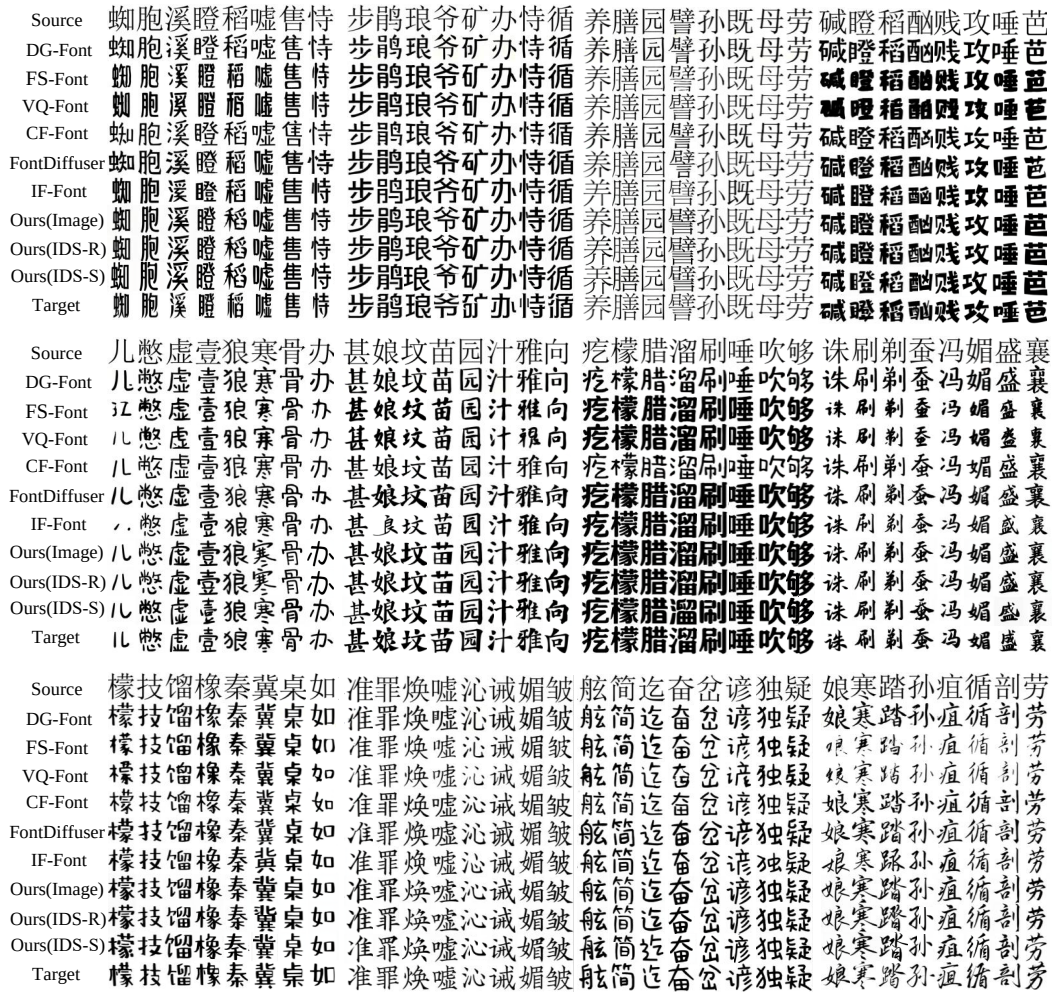

Figure 10: Additional comparison results on UFUC.

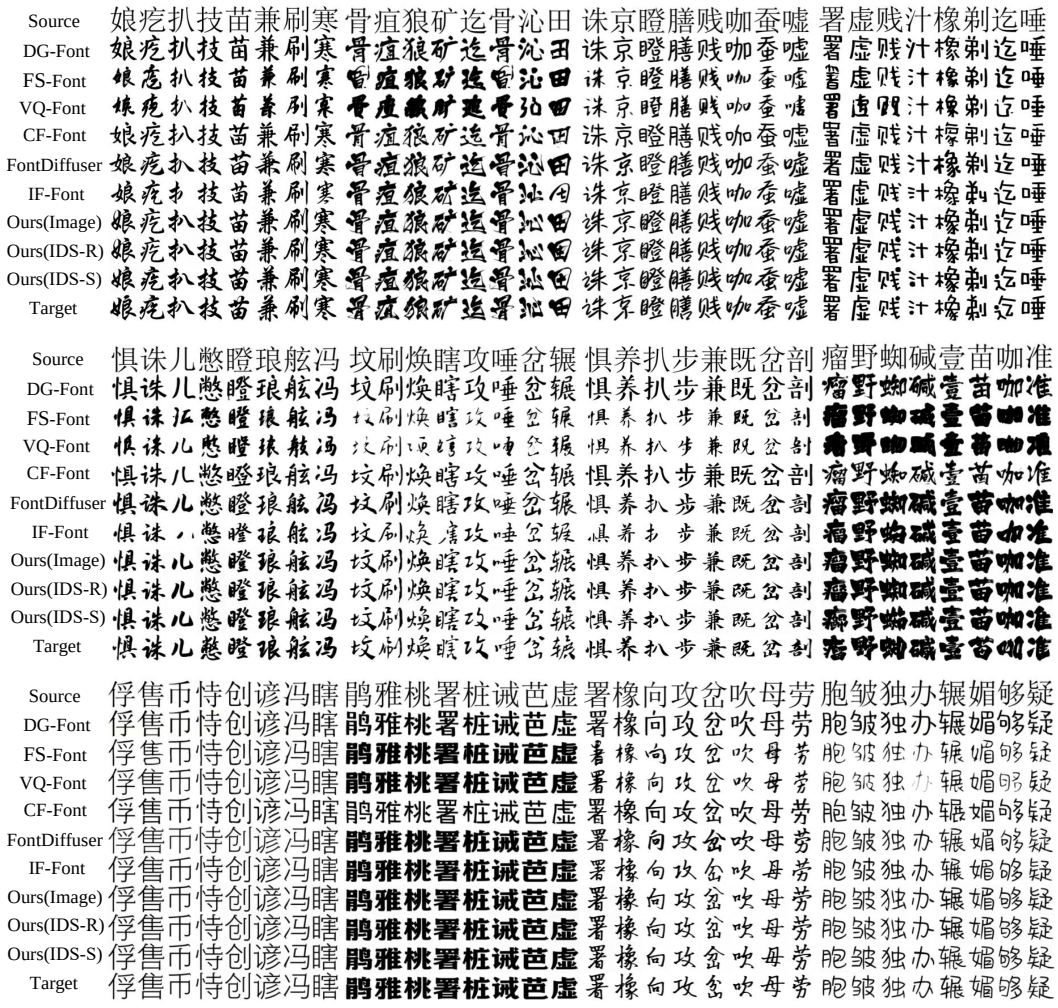

Figure 11: Additional comparison results on SFUC.