Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)
Special Track on AI, the Arts and Creativity

DP-Font: Chinese Calligraphy Font Generation Using Diffusion Model and Physical Information Neural Network

Liguo Zhang[1,2], **Yalong Zhu**[1,2], **Achref Benarab**[1,2],
Yusen Ma[1,2], **Yuxin Dong**[1,2], **Jianguo Sun**[1,2*]
[1]College of Computer Science and Technology, Harbin Engineering University, China
[2]Modeling and Emulation in E-Government National Engineering Laboratory, China
{zhangliguo, zhuyalong, achref.benarab, mayusen, dongyuxin, sunjianguo}@hrbeu.edu.cn

Abstract

As a typical visual art form, Chinese calligraphy has a long history and aesthetic value. However, current methods for generating Chinese fonts still struggle with complex character shapes and lack personalized writing styles. To address these issues, we propose a font generation method for Chinese Calligraphy based on diffusion model incorporating physical information neural network (PINN), which is named DP-Font. Firstly, the multi-attribute guidance is combined to guide the generation process of the diffusion model and introduce the critical constraint of stroke order in Chinese characters, aiming to significantly improve the quality of the generated results. We then incorporate physical constraints into the neural network loss function, utilizing physical equations to provide in-depth guidance and constraints on the learning process. By learning the movement rule of the nib and the diffusion pattern of the ink, DP-Font can generate personalized calligraphy styles. The generated fonts are very close to the calligraphers' works. Compared with existing deep learning-based techniques, DP-Font has made significant progress in enhancing the physical plausibility of the model, generating more realistic and high-quality results.

1 Introduction

Chinese characters are one of the oldest writing systems in the world, and their artistic expression is known as Chinese calligraphy. The calligraphy contains the diverse fonts and styles and combines the practical and aesthetic functions of the writing. The calligraphy creation with multiple styles, such as regular script, running script and cursive script, directly reflects the diversity of Chinese culture [Wang *et al.*, 2023b]. However, beautiful handwriting would require a long time of practice, this is a challenge for everyone's strength and patience. With the development of artificial intelligence, the generation model of Chinese characters has had a profound effect on art and culture. In calligraphy work and layout design, the intelligent generation model can display powerful creativity and flexibility for various fonts.

Figure 1: Illustration of font generation on two styles of Chinese calligraphy fonts using DP-Font. The left and right poems are respectively the five-character quatrains from ancient China. The left one is Gongquan Liu's style, while the right one is Zhenqing Yan's style. Gongquan Liu (778-865 AD) and Zhenqing Yan (709-784 AD) were famous Chinese calligraphers in Tang dynasty.

At present the generation strategies of Chinese character fonts can be approximately divided into two categories: graphics-based and deep learning-based methods. Chinese character generation based on graphics depends largely on strokes extraction and reconstruct influenced by prior knowledge. These kinds of methods are practical, but they hold a significant computational overhead. Additionally, the generation method usually only focuses on the local structure of the character, ignoring the overall calligraphy style. The development of deep learning has changed this situation. The generative adversarial network (GAN) can learn global feature of Chinese character and create (or improve) the existing fonts. For example, Zi2zi [Tian, 2017] is the first generation model for Chinese characters that employed GAN. Afterwards, various GAN-based generation methods have been proposed such as MX-Font [Park *et al.*, 2021], DG-Font [Xie *et al.*, 2021], and CFGAN [Hassan *et al.*, 2023]. Similar to the end-to-end frameworks, the aforementioned methods can capture the global features of the character, but the current solutions cannot capture multiple local styles of Chinese character fonts and predict the global structure of an unknown character. Many algorithms simply adopt the GAN-based

*Corresponding author

image style transfer mechanism. However, these methods introduce a series of problems, such as an unstable training process for the model, inexact font, and limited fidelity of the character. In addition, the overreliance on source domain knowledge will result in over-fitting of the model in the target domain, particularly when large difference exists between both domains, involving a large number of Chinese characters [Han, 2012] (about 60,000) and multitude of fonts. In recent years, the diffusion model [Yoon *et al.*, 2020] has made remarkable achievements in image generation [Rombach *et al.*, 2022]. Diff-Font [He *et al.*, 2022] firstly introduces the diffusion model into font generation task. It utilizes the ordinal number of strokes as a constraint, which significantly improves the quality of the font. However, the generation effect is very limited regarding the spatial distribution of Chinese characters. It is well-known that the rules for handwriting Chinese characters have undergone a long evolution affected by various factors, such as the movement of the nib and the ink spread on the paper. Therefore, we hold the opinion that modeling physical law should be conducive to the diffusion model for font generation. In fact, the traditional diffusion model cannot capture the complex relationships of Chinese characters among local strokes and global structure.

To overcome these problems, the PINN [Von Rueden *et al.*, 2021] is a good option for capturing constraints between strokes and the structure of Chinese characters. PINN can combine the representation capacity of neural networks with the law of physics equation. If the physical constraints are introduced to the loss function, the neural network will improve the font generation model, and thus, the generated results will better satisfy the handwriting rules. In our work, PINN is a powerful tool to generate multiple fonts, due to its sensitivity to physical laws and the flexibility of the neural networks. The generated calligraphy works are shown in Figure 1, where the background with texture is readily available (it is not generated). Thus, it can be seen that the generated Chinese characters by DP-Font have a more genuine and natural vision effect.

To summarize, the primary contributions of this paper are described as follows. Initially, we introduce a novel method for the font generation of Chinese character based on the diffusion model and PINN. The stroke order of Chinese characters is incorporated as one of the constraints to enhance the generation of complex character shapes. Subsequently, physical constraints are incorporated into the neural network loss function. By combining it with physical equations, the learning process is guided and constrained to augment the physical plausibility of the model. Compared to other deep learning-based approaches, the proposed method yields more realistic and higher-quality generative results.

The rest of the paper is organized as follows. In section 2, we briefly summarizes the related work. The details of the proposed DP-Font is described in section3. Section 4 presents the results of comparison experiments and ablation experiments involving Chinese font generation. Finally, the conclusion is drawn in section 5.

2 Related Work

2.1 Font Generation

Nowadays a lot of font generators depend on reference images to produce the fontlib including thousands of characters. Font generation methods are mostly based on style transfer for image-to-image, the core of which is learning the mapping between the input image and the target image. In the learning process, prior information can be applied to the labels of the training data, thereby enhancing both quality and diversity. Hence, early font generation methods mostly relied on prior knowledge, such as the characters with new-style font, to adjust the models. For example, Zi2zi [Tian, 2017] and Rewrite [Tian, 2016] utilize the one-hot encoding to perform supervised learning of GAN and generate new fonts. EMD [Zhang *et al.*, 2018] and SA-VAE [Sun *et al.*, 2017] resolve the potential features into content-related and style-related components to capture various characteristics of Chinese characters, with SA-VAE additionally incorporating prior knowledge. The component conditions can be employed to constrain the encoders in MX-Font [Park *et al.*, 2021] and CG-GAN [Kong *et al.*, 2022] and extract features from reference images. In addition, the unsupervised learning is also used for font generation, for example, DG-Font [Xie *et al.*, 2021] is a deformable generating network that can produce more complex fonts. Last year, some scholars studied the hybrid model to refine the font style. CF-Font [Wang *et al.*, 2023a] applies the content fusion module and iterative style-vector refinement to decouple the content and style of characters. CFGAN [Hassan *et al.*, 2023] and Diff-Font [He *et al.*, 2022] generate the realistic font images based on GAN and diffusion model, respectively. However, the current methods are difficult to satisfied both glyph reconstruction (without error and blurriness) and style fusion (with style of reference font).

2.2 Diffusion Models

Within the domain of generative models, diffusion models stand out by employing a two-phased approach: an initial diffusion process that methodically degrades data fidelity through Gaussian noise application, followed by a reverse diffusion process that recovers the original data structure, with the aid of Markov chains to facilitate this intricate transformation. Jascha et al. [Sohl-Dickstein *et al.*, 2015] first clarified the concept of diffusion probability model.

The denoising diffusion probability model [Ho *et al.*, 2020] improved the theory and applied UNet to predict the additional noise in the image at each diffusion time step. Dhariwal et al. [Dhariwal and Nichol, 2021] propose a classifier guidance mechanism that utilizes pre-trained classifiers to provide gradients as guidance for generating images of target classes. In contrast, the classifier-free diffusion guidance technique proposed by Ho et al. [Ho and Salimans, 2022] eliminates the need for classifiers by jointly training conditional and unconditional diffusion models. This approach achieves a balance between conditional and unconditional fractional functions through a linear combination, ensuring their integration. The denoising diffusion implicit model (DDIM) proposed by Song et al. [Song *et al.*, 2020] extends Ho's approach to handle non-Markovian cases. It provides enhanced predictive ac-

curacy with larger step length, substantially decreasing the number of required sampling steps from dozens to just one. Glide [Nichol *et al.*, 2021], dalle2 [Ramesh *et al.*, 2022], and stable diffusion employ pre-trained text encoders to produce semantic latent spaces, achieving remarkable performance in text-to-image tasks.

2.3 Physical Information Neural Networks

PINN integrates principles from physical systems with deep learning techniques to enhance data processing performance and pattern recognition accuracy [Raissi *et al.*, 2019]. In contrast to conventional neural networks, PINN effectively integrates prior knowledge of data structures to grasp intricate patterns. Proficient at solving partial differential equations, PINN can be adapted for diverse applications, encompassing Bayesian methodologies, physics-informed Generative Adversarial Networks (GANs) for stochastic differential equations [Cai *et al.*, 2021], and data-driven exploration of physical equations in intricate systems. PINN's capacity to grasp extended dependencies and global context in image processing provides inventive solutions to intricate issues. The mechanism of PINN fuses the representation learning capabilities of neural networks with an understanding of the underlying physics equations. By incorporating physical constraints into the loss function, PINN can enhance the font generation model, ensuring that the generated outcomes conform to the principles of handwriting movement.

3 Methodology

In this section, we introduce DP-Font in detail. First, we describe the framework of the model by adopting different generation strategies, including content, stroke order, and style attributes (Section 3.1). Next, we design the training process through a multi-attribute conditional diffusion model (Section 3.2). Finally, we embed the constraints of the PINN into the diffusion model (Section 3.3).

3.1 Framework of DP-Font

As basic of diffusion model, the denoising diffusion probability model (DDPM) consists of two main components: the forward diffusion process and the reverse denoising process. Illustrated in Figure 2 (**Sample**), the forward diffusion process is characterized by a Markovian data corruption sequence, expressed as $q(x_t|x_{t-1})$. This phase involves the gradual infusion of Gaussian noise into each data sample, methodically transforming the original data distribution into a canonical Gaussian distribution. In contrast, the reverse denoising process involves reconstructing the data from its noisy state, denoted as x_i (where $i = 0, 1, \ldots, T$). Here, $p_\theta(\cdot)$ represents the conditional distribution of x_i, effectively guiding the denoising trajectory.

The reverse process, denoted as $p_\theta(x_{t-1}|x_t)$, epitomizes a denoising mechanism where samples are sequentially retrieved from the standard Gaussian distribution. In each iteration, we gradually reduce a little Gaussian noise, so that the samples gradually approach the real data distribution, especially the distribution of standard Chinese characters. This progressive refinement culminates in acquiring samples that are representative of their true data distribution. The aim of this procedure is to yield samples that align closely with the real data distribution, thus enabling effective data restoration and denoising.

The DP-Font framework is described in Figure 2, consisting of two main components. The first is the character attribute encoding module, responsible for transforming character features (including content, stroke order, and style) into the latent variable z. The second component is the DDPM, which utilizes the latent variable z to generate the required images. The DDPM generates character images from Gaussian noise. The character attribute encoder is specifically designed to parse three features of character images: content (denoted as c), stroke order (denoted as stk), and style (denoted as sty). Within the encoder f, content, stroke order, and style are encoded into the latent variable z, where $z = f\left(f_1\left(c\right), f_2\left(stk\right), f_3\left(sty\right)\right)$. Style and content features are extracted with the style encoder, followed by the pre-training of the encoder. The parameters of this encoder are then frozen during the training of our diffusion model.

Unlike traditional image font generation techniques, our approach treats characters with different contents as distinct categories. For stroke order, in accordance with the "Standard for the Arrangement of Chinese Character Strokes" issued by the Ministry of Education of the People's Republic of China on March 1, 2021, there are five most basic types of strokes in Chinese characters, as illustrated in Figure 3a. We assign a unique encoding to each stroke type, then each character is encoded into a 36-dimensional vector (the maximum stroke count in commonly used Chinese characters is 36). Each dimension of the vector represents the encoding corresponding to the respective basic stroke it contains, as shown in Figure 3b. Compared to one-bit and stroke count encoding, this stroke order encoding more effectively represents the stroke attributes of characters. Subsequently, the stroke order vector is expanded to align with the dimensions of content and style embedding. Through this approach, we obtain an attribute representation of character images, subsequently concatenating them to form the conditional z for the subsequent training of the conditional diffusion model.

This encoding approach simplifies intricate Chinese characters into sequences of numbers, making them more suitable for computation and processing. Consequently, it facilitates a more profound understanding of the structure and stroke relationships of Chinese characters by neural network. Through distinctly indicating the type and order of each stroke, this method alleviates ambiguities in the structure of Chinese characters, thereby enhancing learning efficiency, especially in scenarios involving complex or similar characters.

3.2 Multi-Attributes Conditional Diffusion Model

The forward process of DDPM involves a Markov chain-based denoising procedure. Assuming T is the total number of noise-adding steps, the initial data sample distribution is $x_0 \sim q(x_0)$. At each step t of the forward process, Gaussian noise with specific mean and standard deviation values is added, as shown in Eq. (1):

$$\begin{cases} q\left(x_t \mid x_{t-1}\right) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}\right) \\ q\left(x_{1:T} \mid x_0\right) = \prod_{t=1}^{T} q\left(x_t \mid x_{t-1}\right) \end{cases} \tag{1}$$

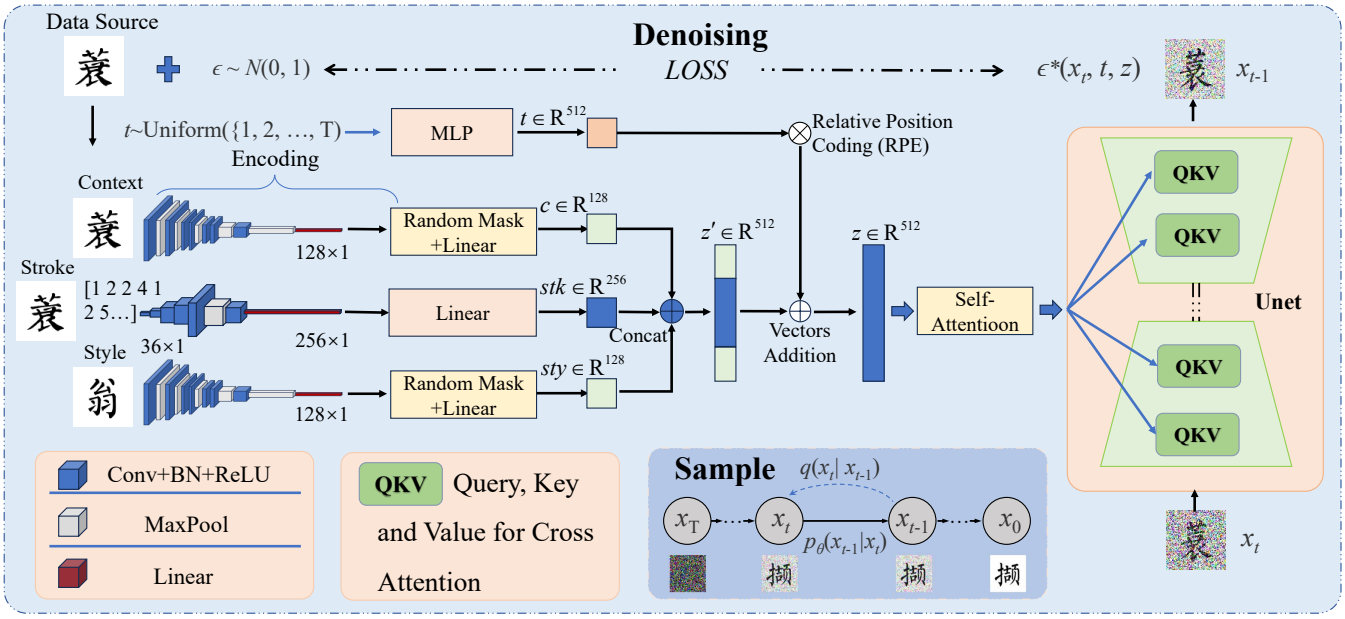
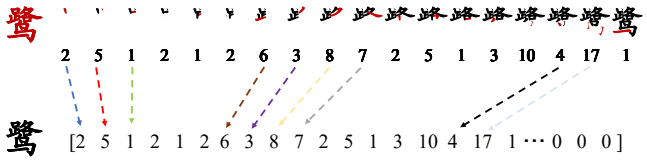


Figure 2: Framework of DP-Font. **Denoising**: The noise step t and the noisy Chinese character sequence x_t (including the content of the character, stroke order, and style features) conditioned on z at that noise step are fed into the diffusion model. We utilize the latent variable z as a condition for training the DDPM, and random masks in the processing pipeline for Chinese character content and style features to assist in classifier-free training of the model. **Sample**: At each step t , we predict ϵ using the denoising process based on the corresponding conditions, and then add noise to the denoised step x_{t-1} using the diffusion process. This process is repeated from $t=T$ down to $t=0$.

Strokes	heng(一)	shu(丨)	pie(丿)	dian(丶)	zhe(乛)
Encoding	1	2	3	4	5

(a)



(b)

Figure 3: Illustration of stroke order. (a) Five most basic strokes of Chinese, (b) Illustration of character stroke order encoding.

Here, x_t is the data with added noise at time t, and β_t is the manually set noise addition parameter at t. The noise schedule is β_1, β_2,..., β_T with $0<\beta_1<\beta_2<\ldots<\beta_T<1$. The noise addition can be calculated as:

$$x_t = \sqrt{1-\beta_t} x_t - 1 + \sqrt{\beta_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}) \tag{2}$$

Defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$. Under Markov's assumptions, the Eq. (2) iteratively becomes:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}) \tag{3}$$

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right) \tag{4}$$

From Eqs. (1)-(4), for a given x_0 and noise schedule, x_t at any step can be obtained directly. With a sufficiently large T, the final noise addition result x_T approximates a Gaussian distribution, $q(x_T \mid x_0) \sim \mathcal{N}(0, \mathbf{I})$, where I is the identity matrix. In image generation, this forward process transforms the original image into a noisy one.

The inverse process gradually eliminates noise to reconstruct data and is employed for data generation post-training. In the DDPM, this process also follows a Markov chain. Assuming the conditional probability distribution $p_\theta(x_{t-1}|x_t, z)$ is accurately determined at each step t, iterative sampling in the reverse direction accomplishes the generation task. However, directly finding $q(x_0, z)$ is impractical, so a neural network parameterized by θ approximates its distribution. It's assumed that $p_\theta(x_{t-1}|x_t, z)$ follows a Gaussian distribution, with mean μ_θ and variance $\sum\theta$ taking x_t, t, and z as input parameters:

$$p_\theta(x_{t-1} \mid x_t) := \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t, z), \sigma^2\mathbf{I}\right) \tag{5}$$

Here,$z = f(f_1(c), f_2(stk), f_3(sty))$ encodes content, stroke order, and style attributes into feature vectors.

To simplify computation and ease neural network training, variance σ^2 is set as a time-dependent constant, not involved in training. Thus, training focuses on the mean μ:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_{t-1}}}\epsilon_\theta(x_t, t, z)\right) \tag{6}$$

Here, ϵ_θ represents noise prediction made by the diffusion model, with UNet learning to predict ϵ_θ. The loss function, Mean Square Error (MSE), quantifies the difference between the actual and predicted noise:

$$\mathcal{L}_{\text{simple}} = \mathbf{E}_{x_0,\epsilon,z}\left[\|\epsilon - \epsilon_\theta(x_t, t, z)\|^2\right] \tag{7}$$

As the algorithm is based on generating Chinese characters under controlled conditions, these conditions can encompass

not only the stroke order stk, but also the content c or the style sty, among others. As illustrated in Figure 2, we introduce random masking to the content c and style sty features within the processing pipeline for classifier-free learning. This approach enables precise control over varying conditions. Subsequently, the generation of Chinese characters is facilitated by a classifier-free guidance, achieved by combining the conditional model $\epsilon_\theta\left(x_t, t, z_1\right)$, $z_1 = (c, stk, sty)^T$, with the unconditional model $\epsilon_\theta\left(x_t, t, z_2\right)$, $z_2 = (c, \varnothing, \varnothing)^T$ where no style and stroke order conditions are added, during the training process. This is expressed in the formula:

$$\hat{\epsilon_\theta} = \omega\epsilon_\theta(x_t, t, z_1) + (1-\omega)\epsilon_\theta(x_t, t, z_2), \quad \omega \in [0, 1] \quad (8)$$

The formula represents the predictive blending of both models to generate Chinese characters However, this approach encounters challenges such as requiring numerous iterations and involving complex calculations, leading to limited control over the diffusion model in the generation process. To overcome these issues, PINN is integrated into the diffusion model (section 3.3). PINN integrates physical laws (e.g., diffusion equations) to guide learning, reducing iterations and simplifying computations, thus enhancing the diffusion model's efficiency. Additionally, the physical properties of the diffusion equation help the model to simulate font characteristics more realistically, reducing excessive smoothing and retaining more natural features.

3.3 Incorporating PINN into Diffusion Model

As an advanced deep learning framework, the PINN integrates neural networks with principles of physics. In the framework, neural networks are assigned the dual responsibility of not only learning data features but also incorporating physics equations to guide and constrain the learning process. This methodology is particularly advantageous in scenarios where the physical process is intricate or when acquiring extensive data is challenging.

In the context of font generation, we employ diffusion equations akin to those in thermal diffusion or fluid dynamics to emulate the ink diffusion effect on paper, as delineated in Eq.(9):

$$\frac{\partial u}{\partial t} = D\nabla^2 u \quad (9)$$

Here, $u\left(x, y, t\right)$ represents the ink concentration at the position (x, y) and time t, D is the diffusion coefficient, and ∇^2 is the Laplace operator, indicating the second spatial derivative. The PINN losses are computed using Mean Square Error (MSE):

$$\mathcal{L}_{\text{PINN}} = \sum_{(x,y,t)} \left\| \frac{\partial \hat{u}}{\partial t} - D\nabla^2 \hat{u} \right\|^2 \quad (10)$$

This is combined with the diffusion loss $\mathcal{L}_{\text{simple}}$ to form the total loss function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{simple}} + \lambda\mathcal{L}_{\text{PINN}} \quad (11)$$

In this equation, λ is a weighting coefficient that balances the significance of different loss terms. In our experiments (section 4), λ is empirically set to 1. Through this approach, the diffusion model learns not just the data characteristics but also solutions that comply with the physical diffusion process. This not only enhances the physical authenticity of the model but also elevates its precision in modeling complex phenomena.

4 Experimental Results

4.1 Dataset and Experimental Setting

In the process of font generation, we have compiled a dataset containing a variety of diverse fonts. Specifically, the dataset includes 100 fonts, with each font containing 7,905 characters. These characters are sourced from the 2021 General Standard Chinese Character Stroke Order Standard issued by the National Language Commission of the Ministry of Education in the People's Republic of China (PRC), ensuring coverage of common Chinese characters. To accurately assess the performance of the font recognition algorithm, a distinct test set has been created. The test set consists of 10 fonts and incorporates 200 characters distinct from those in the main dataset. The characters in the test set were intentionally chosen due to their intricate structures and multiple strokes. All characters in our dataset are resized to a consistent 80×80 pixels (image size). This normalization is crucial to eliminate size-dependent variables, facilitating comparison of performance with other methods.In our experiments, DP-Font is implemented using Pytorch 3.6 running on GeForce RTX 3090 graphics.

4.2 Evaluation Metrics

To effectively compare our font generation method with leading techniques, we utilize four evaluation metrics: Structural Similarity Index (SSIM) [Wang *et al.*, 2004], Root Mean Square Error (RMSE), Peak Signal-to-Noise Ratio (PSNR), and Frechet Inception Distance (FID) [Heusel *et al.*, 2017]. SSIM assesses image similarity based on brightness, contrast, and structure, reflecting the perception of the human visual system. RMSE calculates the pixel-level similarity between two images, providing a measure of precision. PSNR is a metric used in image processing to quantify the quality of a reconstructed image compared to its original version, with higher values indicating greater similarity and better image quality. FID measures the distributional difference between generated and real images, focusing on texture, pattern, and style. These metrics offer a comprehensive framework for evaluating font generation quality, enabling us to assess the performance of our method against existing state-of-the-art techniques.

4.3 Comparison Experiments

The proposed DP-Font undergoes testing for comparison with five current font generation algorithms, including FUNIT [Liu *et al.*, 2019], SC-Font [Jiang *et al.*, 2019], MX-Font [Park *et al.*, 2021], DG-Font [Xie *et al.*, 2021] and Diff-Font [He *et al.*, 2022]. (1) FUNIT: A method for image-to-image translation which is able to use limited tag data during training and seamlessly handle the conversion between different

Figure 4: The results of generating simple and complex Chinese characters using DP-Font and other five methods. Left ten columns are simplified Chinese characters, including regular script and running script, and right ten columns are complex Chinese characters, including Song typeface and artistic boldface. The red boxes and circles respectively point out the global and local mistakes of generated characters. Qualitatively, the generated results by DP-Font are nearest the target fonts and little wrongly written characters.

fields (2) SC-Font: Using CNN model to learn how to transfer the writing trajectory with separated strokes from a reference font style to a target font style. (3) MX-Font: MX-Font is a multi-localized expert network for Few-Shot font generation (4) DG-Font: DG-Font introduces a feature deformation skip connection to predict pairs of displacement maps, and apply deformable convolution to the low-level feature maps from the content encoder. (5) Diff-Font: Diff-Font uses conditional generative diffusion model for font generation, which differs from traditional image-to-image translation methods.

We train these algorithms using the datasets described in section 4.1, restricting to a single reference font for generation. To evaluate these methods, we utilize the Song font, a widely used standard font in Chinese character generation tasks, as the source font.

Methods	SSIM↑	RMSE↓	PSNR↓	FID↓
FUNIT	0.681	0.320	12.40	25.52
SC-Font	0.694	0.298	9.87	25.78
MX-Font	0.689	0.312	11.95	28.46
DG-Font	0.712	0.295	9.24	27.56
Diff-Font	0.720	0.283	8.62	24.30
DP-Font	**0.735**	**0.279**	**8.34**	**22.34**

Table 1: Quantitative comparison for generation results of Chinese characters with other methods.

Quantitative Evaluation. Table 1 present the quantitative comparison results of DP-Font with other current generation methods. It is evident that our method outperforms others in all metrics. DP-Font achieves the highest SSIM at 0.735, signifying the best image quality, it also attains the lowest scores in RMSE (0.279), PSNR (8.34), and FID (22.34), indicating that it is the most accurate and efficient in font generation among the compared methods. ↑ denotes that a larger value typically corresponds to better performance, while ↓ signifies that a smaller value usually indicates better performance.

DP-Font has demonstrated optimal performance across SSIM, RMSE, PSNR, and FID evaluation metrics. A particularly significant observation is the notable 8.48 % improvement in our method's performance in the FID metric over Diff-Font, highlighting its superiority.

Qualitative Evaluation. The qualitative results are illustrated in Figure 4. We selected four fonts, including two simple and two complex ones, along with 20 Chinese characters. This set comprises ten characters with simple strokes and structures and ten characters with complex strokes and structures. These characters were generated using the six training completion methods, respectively. It can be seen that FUNIT exhibits stroke errors for the majority of generated Chinese characters and fonts, struggling to maintain the integrity of characters. SC-Font shows issues with missing and redundant strokes, affecting the overall character structure. MX-Font maintains the general shape but often lacks clarity and definiteness in the generated characters. DG-Font demonstrates competence in simpler environments but struggles to capture complex details in more challenging tasks. Diff-Font Demonstrates competence in simpler environments but struggles to capture complex details in more challenging tasks. The proposed DP-Font method consistently outperforms other approaches in all cases, indicating its robustness and effectiveness in generating high-quality Chinese font style transformations.

In addition to the fonts of standard library, DP-Font can also learn and imitate the handwriting of multiple calligraphers, as shown in Figure 1 and Figure 5. In Figure 1 left and right two poems respectively are written in Gongquan Liu's and Zhenqing Yan's styles. Gongquan Liu (778-865 AD) and Zhenqing Yan (709-784 AD) were famous Chinese calligraphers in Tang dynasty. As target fonts, Figure 5 (a) and (c) are two kinds of handwriting for a same poem. The originators of two fonts respectively are Zhengming Wen (1470-

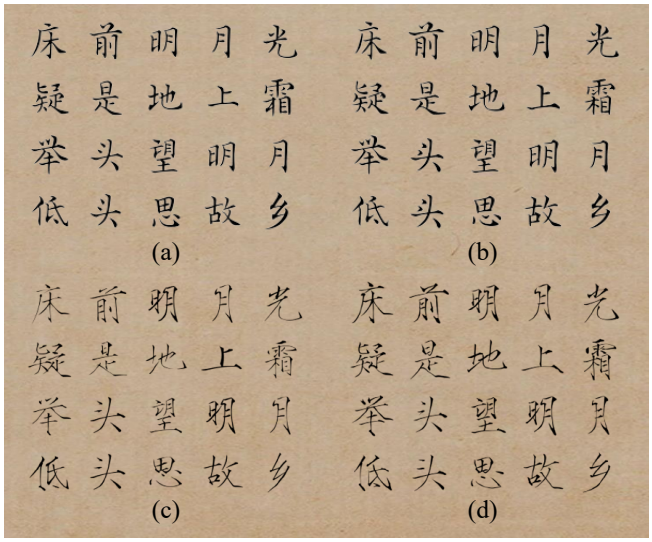


Figure 5: Illustration of calligraphic works with two styles. (a) and (c) are the target fonts of Chinese calligraphy with Zhengming Wen's and Ji Zhao's styles. (b) and (d) are the generated results from DP-Font.

1559 AD, was a Chinese painter, calligrapher, and poet during the Ming dynasty) and Ji Zhao (1082-1135 AD, was the eighth emperor and very well-known artist of the Song dynasty of China). Figure 5(b) and (d) are calligraphy works generated by DP-Font. Obviously, DP-Font can learn writing rules from different people's handwriting, and the generated Chinese characters have very realistic style and details.

4.4 Ablation Study

In this section, we conduct an ablation study to discuss the effectiveness of stroke order encoding and explore the impact of ω (guidance scales). Figure 6 shows the results of ablation studies under different stroke conditions. The last row is the ground truth, and the first to fourth rows are the results of DP-Font with no stroke condition, stroke encoded with one bit, stroke encoded with stroke count, and stroke encoded with stroke order, respectively.

We trained four DP-Fonts on the small dataset, one without using stroke conditions, one using stroke encoded with one bit, one using stroke encoded with stroke count, and one using stroke encoded with stroke order. As shown in Figure 6, when generating characters with difficult structures, the DP-Font without explicit encoding of stroke order and quantity may generate characters with stroke errors. The results in Table 1 show that adding stroke order encoding implicitly includes the number of strokes, so the generated Chinese characters have higher quality and better performance than the Diff-font.

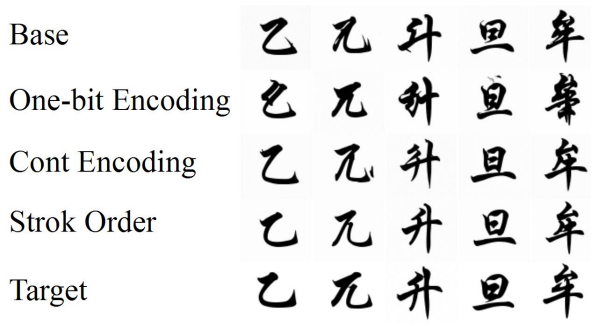


Figure 6: Qualitative results of ablation studies using different stroke conditions.

Guidance Scale	SSIM↑	RMSE↓	PSNR↓	FID↓
$\omega = 0.2$	0.723	0.292	8.70	23.12
$\omega = 0.4$	0.728	0.283	8.54	22.70
$\omega = 0.6$	**0.735**	**0.279**	**8.34**	22.34
$\omega = 0.8$	0.732	0.287	8.61	**22.19**

Table 2: Impact of different guidance scale on experimental results.

Methods	SSIM↑	RMSE↓	PSNR↓	FID↓
RAW	0.699	0.332	11.64	24.91
RAW+S_O	0.701	0.314	11.21	24.06
RAW+PINN	0.718	0.309	10.66	23.57
RAW+S_O +PINN	**0.735**	**0.279**	**8.34**	**22.34**

Table 3: Evaluation scores of various methods in ablation experiments.

The impact of guidance scale. By setting different conditional scale, ω , which is defined in Eq.(8), we further discussed the impact of conditional and unconditional on generation. The contrast experiments are conducted on the test set of the large dataset mentioned in Section 4.1. As shown in Table 2, we found that setting $\omega = 0.6$ can achieve the best quality of generated images.

To further discuss the validity of stroke order encoding and PINN, we respectively train basic diffusion model (RAW) and its variants (RAW+S_O, RAW+PINN, and RAW+S_O+PINN=DP-Font). The RAW method does not incorporate the stroke order coding, RAW+S_O adds the feature control condition of stroke order on top of RAW, and RAW +PINN adds the PINN loss constraint on top of RAW. As depicted in Table 3, the quantitative results of all evaluation metrics demonstrate improvement when adding the stroke order condition with PINN.

5 Conclusion

In this paper, we propose DP-Font, a font generation method for Chinese Calligraphy based on diffusion model incorporated PINN. DP-Font adopts a multi-attribute approach to guide the generation of diffusion models, incorporating stroke order as a constraint to enhance the generated character font results. Additionally, we integrate physical constraints into the neural network loss term and incorporating physical equations to guide and constrain the learning process, enhancing the model's physical rationality. DP-Font produces more realistic and higher-quality generation results compared to other deep learning-based methods. Experimental results confirm the superiority of the proposed method among similar techniques.

Acknowledgments

This work is supported by National Key R&D Program of China(2022YFB4400700).

References

[Cai *et al.*, 2021] Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6):060801, 2021.

[Dhariwal and Nichol, 2021] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[Han, 2012] Jiantang Han. *Chinese characters*. Cambridge University Press, 2012.

[Hassan *et al.*, 2023] Ammar Ul Hassan, Irfanullah Memon, and Jaeyoung Choi. Real-time high quality font generation with conditional font gan. *Expert Systems with Applications*, 213:118907, 2023.

[He *et al.*, 2022] 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. *arXiv preprint arXiv:2212.05895*, 2022.

[Heusel *et al.*, 2017] 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.

[Ho and Salimans, 2022] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[Jiang *et al.*, 2019] Yue Jiang, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. Scfont: Structure-guided chinese font generation via deep stacked networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4015–4022, 2019.

[Kong *et al.*, 2022] 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*, pages 13482–13491, 2022.

[Liu *et al.*, 2019] 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*, pages 10551–10560, 2019.

[Nichol *et al.*, 2021] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[Park *et al.*, 2021] 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*, pages 13900–13909, 2021.

[Raissi *et al.*, 2019] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[Ramesh *et al.*, 2022] 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.

[Rombach *et al.*, 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*, pages 10684–10695, 2022.

[Sohl-Dickstein *et al.*, 2015] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[Song *et al.*, 2020] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[Sun *et al.*, 2017] Danyang Sun, Tongzheng Ren, Chongxun Li, Hang Su, and Jun Zhu. Learning to write stylized chinese characters by reading a handful of examples. *arXiv preprint arXiv:1712.06424*, 2017.

[Tian, 2016] Yuchen Tian. Rewrite: Neural style transfer for chinese fonts. *https://github.com/kaonashi-tyc/rewrite*, 2016.

[Tian, 2017] Yuchen Tian. Zi2zi: Master chinese calligraphy with conditional adversarial networks. *https://github.com/kaonashi-tyc/zi2zi*, 2017.

[Von Rueden *et al.*, 2021] Laura Von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, et al. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2021.

[Wang *et al.*, 2004] 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.

[Wang *et al.*, 2023a] 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*, pages 1858–1867, 2023.

[Wang *et al.*, 2023b] Luya Wang, Nor Azlin Hamidon, et al. Aesthetic and value study of inscriptions from the perspective of calligraphy. *Art and Performance Letters*, 4(11):44–49, 2023.

[Xie *et al.*, 2021] 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*, pages 5130–5140, 2021.

[Yoon *et al.*, 2020] Youngwoo Yoon, Bok Cha, Joo-Haeng Lee, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. Speech gesture generation from the trimodal context of text, audio, and speaker identity. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020.

[Zhang *et al.*, 2018] Yexun Zhang, Ya Zhang, and Wenbin Cai. Separating style and content for generalized style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8447–8455, 2018.