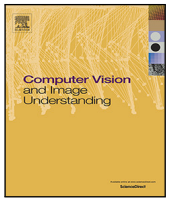
Computer Vision and Image Understanding 245 (2024) 104043

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu



Font transformer for few-shot font generation

Xu Chen, Lei Wu *, Yongliang Su, Lei Meng, Xiangxu Meng

School of Software, Shandong University, Jinan, 250101, China

ARTICLE INFO

Communicated by Nicu Sebe

MSC:
41A05
41A10
65D05
65D17

Keywords:
Font generation
Transformer
Few-shot learning
Image generation

ABSTRACT

Automatic font generation is of great benefit to improving the efficiency of font designers. Few-shot font generation aims to generate new fonts from a few reference samples, and has recently attracted a lot of attention from researchers. This is valuable but challenging, especially for ideograms with high diversity and complex structures. Existing models based on convolutional neural networks (CNNs) struggle to generate glyphs with accurate font style and stroke details in the few-shot setting. This paper proposes the TransFont, exploiting the long-range dependency modeling ability of the Vision Transformer (ViT) for few-shot font generation. For the first time, we empirically show that the ViT is better at glyph image generation than CNNs. Furthermore, based on the observation of the high redundancy in the glyph feature map, we introduce the glyph self-attention module for mitigating the quadratic computational and memory complexity of the pixel-level glyph image generation, along with several new techniques, i.e., multi-head multiple sampling, yz axis convolution, and approximate relative position bias. Extensive experiments on two Chinese font libraries show the superiority of our method over existing CNN-based font generation models, the proposed TransFont generates glyph images with more accurate font style and stroke details.

1. Introduction

At present, the creation of font libraries relies heavily on the manual work of font designers. The purpose of automatic font generation is to let the font designer design only part of the glyph, and then automatically generate all the remaining glyphs in a font, which can reduce the workload of the designer, improve the efficiency of font library production. On the other hand, we hope to rely on as few reference samples as possible, that is, few-shot font generation, so as to further improve the efficiency of font library creation. Compared with the non-few-shot method (Huang et al., 2020; Jiang et al., 2017, 2019; Lyu et al., 2017; Wu et al., 2020a,b; Chang et al., 2018; Gao and Wu, 2020; Wen et al., 2021; Zeng et al., 2021), few-shot font generation (Chen et al., 2021b; Kong et al., 2022; Park et al., 2020, 2021; Tang et al., 2022) can be applied in a wider range of scenarios, e.g., historical handwriting repair, handwriting font imitation, etc. Few-shot font generation has attracted a lot of attention from researchers recently.

Few-shot font generation is challenging, especially for ideograms with high diversity and complex structure, e.g., there are 70,244 Chinese characters in the official standard GB18030-2005, many of which with complex radicals. Existing models based on CNNs struggle to generate glyphs with accurate font style and stroke details in the few-shot setting. We argue that this is due to the limited ability in shape recognition (Geirhos et al., 2018). Recent studies have shown that the ViT (Dosovitskiy et al., 2020) has strong transferability in few-shot learning (Naseer et al., 2021) and performs better than CNNs on shape recognition (Naseer et al., 2021; Tuli et al., 2021). Note that the glyph image naturally expresses shape information. Motivated by this, we developed a transformer-based font generation model, which shows promising results in the few-shot setting.

Image generation tasks usually employ pixel-level tokens for higher generation quality (Zhang et al., 2021; Jiang et al., 2021; Zhao et al., 2021). In this case, the self-attention module suffers from the quadratic computational and memory complexity. We observed that the redundancy of glyph feature maps is high. Existing efficient self-attention methods (Jiang et al., 2021; Xia et al., 2022; Yue et al., 2021; Zhang et al., 2021; Zhao et al., 2021; Zhu et al., 2020) lack consideration for the unique sparsity of glyph images and they impair the long-range dependency modeling ability. We assume that the glyph feature map can be represented by a small number of critical tokens and propose the glyph self-attention module, which dynamically attends to a small number of representative tokens, as shown in Fig. 1(d). Our approach is more flexible and robust than the local window methods (Jiang et al., 2021; Liu et al., 2021) (shown in Fig. 1(b)) and the global fixed anchor method (Zhao et al., 2021) (shown in Fig. 1(c)), which mitigates the quadratic complexity of the self-attention mechanism without compromising the long-range dependency modeling ability for diverse glyphs.

* Corresponding author.
E-mail address: i_lily@sdu.edu.cn (L. Wu).

https://doi.org/10.1016/j.cviu.2024.104043
Received 24 March 2023; Received in revised form 5 May 2024; Accepted 20 May 2024
Available online 24 May 2024

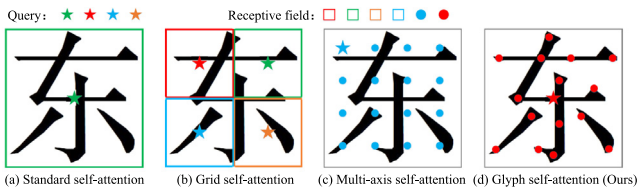


Fig. 1. Comparison of the schematic of different self-attention methods. A pixel as a token in image generation tasks and the solid circle represents a pixel. (a) The standard self-attention (Dosovitskiy et al., 2020) suffers from the quadratic computational and memory complexity. (b) The grid self-attention (Jiang et al., 2021) sacrifice the long-range dependency modeling ability. (c) The multi-axis self-attention lack flexibility for diverse glyphs. (d) Our approach takes into account the long-range dependency and flexible modeling ability for glyph images.

This paper explores a pure transformer-based model for few-shot font generation, the TransFont, consisting of a content encoder, a style encoder, and a decoder. We keep the standard global self-attention (Vaswani et al., 2017) in the low-resolution stage and replace the standard global self-attention with the glyph self-attention in the high-resolution stage. Different from existing transformer-based generative models (Jiang et al., 2021; Zhang et al., 2021; Zhao et al., 2021), where they learn to generate images from random noise, our model learns a mapping from the source font to the target font conditioned on the reference glyphs.

The proposed glyph self-attention module attends to representative tokens by doing spatial sampling in the glyph feature map, and our approach differs from existing methods (Dai et al., 2017; Xia et al., 2022; Yue et al., 2021; Zhu et al., 2020) mainly in three aspects. First, we propose multi-head multiple sampling, which enables different attention heads to learn different representations explicitly. Second, we propose to perform convolution along the other two axes (denoted as y, z axes) of the glyph feature map for predicting sampling coordinates. Since a reasonable assumption is that the conventional x axis convolution cannot capture glyph information when scanning the non-glyph area. Last but not least, we introduce the approximate relative position bias in the glyph self-attention, which extends the relative position bias (Liu et al., 2021) to the high-resolution stage, as we found that the relative position bias enables the model to generate sharper glyph images. The proposed TransFont is evaluated on two Chinese font libraries, FounderType and SinoType. Extensive experiments show the effectiveness and interpretability of the proposed method. In summary, this paper makes the following contributions:

- We propose TransFont, a pure transformer-based model for few-shot font generation, which empirically shows that the ViT is better at glyph image generation than CNNs, thanks to the ability in shape recognition.
- We propose the glyph self-attention module for mitigating the quadratic computational and memory complexity of the self-attention mechanism on pixel-level glyph image generation, introducing several new techniques, i.e., multi-head multiple sampling, yz axis convolution, and approximate relative position bias.
- We view TransFont as a simple but effective transformer baseline for future research, demonstrating its superiority over existing CNN-based font generation models on two challenging Chinese font libraries, FounderType and SinoType. The proposed TransFont generates glyph images with more accurate font style and stroke details.

2. Related works

2.1. Few-shot font generation

Font generation is an image-to-image translation task (Isola et al., 2017; Zhu et al., 2017) where the model learns a mapping from the source font domain to the target font domain. Non-few-shot methods can be divided into supervised (Huang et al., 2020; Jiang et al., 2017, 2019; Lyu et al., 2017; Wu et al., 2020a,b) and unsupervised (Chang et al., 2018; Gao and Wu, 2020; Wen et al., 2021; Zeng et al., 2021), and their performance drops significantly in the few-shot setting. In addition, artistic font generation (Azadi et al., 2018; Gao et al., 2019) is related to font generation, but their models cannot generalize to font generation.

The few-shot methods differ from the non-few-shot methods in two aspects. First, the separation of content and style representations (Sun et al., 2017; Zhang et al., 2018). Second, they rely on a large number of existing fonts to learn font generation capabilities (Chen et al., 2021b; Kong et al., 2022; Park et al., 2020, 2021; Tang et al., 2022). Recently, some methods propose to learn localized style representations by utilizing character radical annotations, i.e., DM-Font (Cha et al., 2020), LF-Font (Park et al., 2020), and MX-Font (Park et al., 2021). DG-Font (Xie et al., 2021) proposes an unsupervised font generation model based on deformable CNNs (Dai et al., 2017). XMP-Font (Liu et al., 2022) proposes a transformer-based cross-modality pre-training method, but its encoder and decoder are based on CNNs. Both the encoder and the decoder are developed based on the transformer in our work.

2.2. Transformer in computer vision

The success of transformers is expanding from natural language processing (Brown et al., 2020; Devlin et al., 2018; Radford et al., 2018, 2019; Vaswani et al., 2017) to computer vision (Carion et al., 2020; Dosovitskiy et al., 2020; Liu et al., 2021). For image generation, one category of methods generates images indirectly in an autoregressive manner (Chen et al., 2020; Child et al., 2019; Esser et al., 2021; Ramesh et al., 2021), and the other directly generates images by using ViT (Jiang et al., 2021; Zhang et al., 2021; Zhao et al., 2021). Our work falls into the latter but differs in that font generation is an image-to-image translation task, whereas (Jiang et al., 2021; Zhang et al., 2021; Zhao et al., 2021) aim to build a transformer-based GANs (Goodfellow et al., 2014). For image-to-image translation, the multi-task learning method of IPT (Chen et al., 2021a) is not suitable for font generation. For style transfer (Gatys et al., 2016), note that the style here refers to the texture style, while the font style is the shape style. Therefore, existing transformer-based style transfer models, i.e., Wu et al. (2021) cannot generalize to font generation.

Recently, some studies (Naseer et al., 2021; Tuli et al., 2021) have shown that the ViT (Dosovitskiy et al., 2020) has intriguing properties compared to CNNs, such as strong ability in shape recognition and strong transferability for few-shot learning. Previously, Geirhos et al. (2018) shows that the ImageNet-trained CNNs are strongly biased towards recognizing textures rather than shapes. Glyph images naturally express shape information, suggesting that the ViT is better at processing glyph images than CNNs. The strong transferability indicates that the ViT performs better than CNNs in few-shot learning, and our proposed TransFont shows that this is indeed the case for few-shot font generation.

2.3. Efficient self-attention modules

Images lead to the quadratic computational and memory complexity of self-attention, and many efforts have been made to mitigate this problem (Dong et al., 2021; Liu et al., 2021; Vaswani et al., 2021; Zhu et al., 2020). Most relevant to our proposed glyph self-attention is deformable attention (Xia et al., 2022; Zhu et al., 2020) and PS-ViT (Yue et al., 2021), and they are inspired by the idea of sampling in deformable convolutional networks (Dai et al., 2017). It is worth emphasizing that our motivation for using sampling is to make the self-attention attends to representative glyph tokens dynamically and sparsely. From another perspective, Xia et al. (2022) and Yue et al.

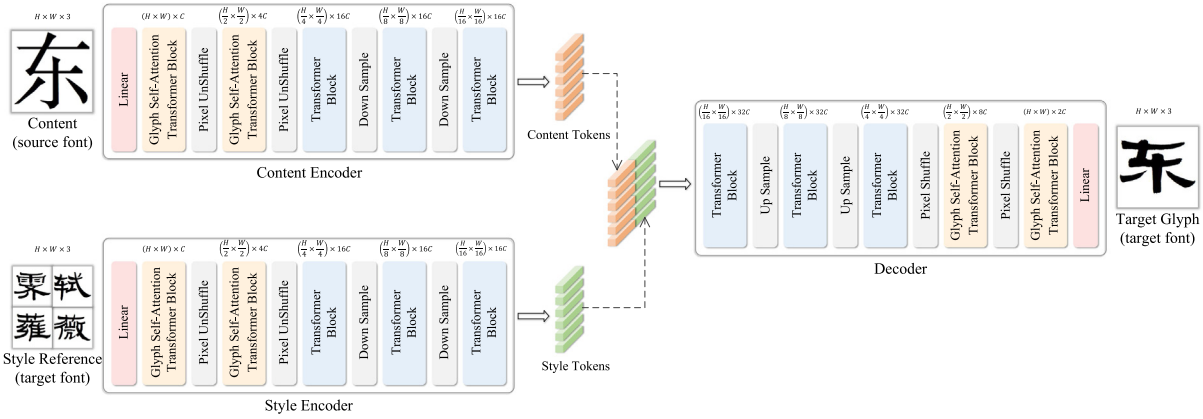


Fig. 2. Overview of the proposed TransFont. The content encoder learns the character content representation from the source font. The style encoder learns the font style representation from target fonts. The content and style tokens are concatenated as glyph tokens, and then the decoder translates the glyph tokens into glyph images.

(2021) are proposed for discriminative tasks, Zhu et al. (2020) is proposed for object detection, but our glyph self-attention is proposed for glyph image generation. In addition, the proposed glyph self-attention is also different from Xia et al. (2022), Yue et al. (2021) and Zhu et al. (2020) in terms of specific technical details, i.e., multi-head multiple sampling, $y\,z$ axis convolution, and approximate relative position bias, which we will introduce in the next section.

3. Approach

3.1. Overview

The proposed TransFont aims at mapping a glyph image of the source font to a glyph image of the target font conditioned on the reference glyph image. As shown in Fig. 2, our model consists of a content encoder, a style encoder, and a decoder. The two encoders have the same architecture. The encoder and decoder are symmetrical. We use PixelShuffle (Shi et al., 2016) in the encoder and PixelUnShuffle in the decoder. The purpose of PixelShuffle is to gradually reduce the embedding dimension of the token, while increasing the resolution of the feature map. The PixelUnShuffle is the inverse operation of PixelShuffle, and its purpose is the opposite of PixelShuffle. As the basic block, the transformer block consists of N transformer encoder layer (Dosovitskiy et al., 2020). In the low-resolution stage, we keep standard self-attention (Vaswani et al., 2017). In the high-resolution stage (resolution higher than 32 × 32), we replace the standard self-attention with the proposed glyph self-attention, which we introduce in section 3.2.

There are only a few samples of new fonts for the model to learn in the few-shot setting, and the style of new fonts is different from the font in the pre-training set in practical scenarios. To ensure generation quality, we generate new fonts by fine-tuning. Our model is trained in supervised fashion, and we adopt L1 loss as the only loss function for pre-training and fine-tuning. Denote the model as G, and the formula is as follows,

$$\mathcal{L}(G) = \mathbb{E}_{c,s,y}[\|G(c,s) - y\|_1],$$

where c is the glyph image from the source font, s is the style reference from target fonts, and y is the ground truth.

We use RGB images rather than grayscale images in our method, because the effect is the same whether the number of channels for the input image is 3 or 1. Taking the encoder in Fig. 2 as an example, if the input image is changed from an RGB image to a grayscale image, the change in the model only occurs in the number of input channels in the first linear layer, and the rest of the model settings do not need to be changed, so the result will not be affected. Our early experiments also showed that there was no difference between using RGB images and grayscale images, thus we follow the existing font generation research and use RGB images.

3.2. Glyph self-attention

Standard Multi-head Self-attention. We first revisit the standard multi-head self-attention in transformer (Vaswani et al., 2017). Denote $x \in \mathbb{R}^{H\times W\times C}$ as the input feature map, where $H\times W$ is the resolution and C is the channel dimension. Before computing multi-head self-attention, we first flatten $x \in \mathbb{R}^{H\times W\times C}$ to $x \in \mathbb{R}^{N\times C}$, where $N = H\times W$. The multi-head self-attention is formulated as

$$Q = xW_q,\ K = xW_k,\ V = xW_v, \tag{1}$$

$$head_i = SoftMax(Q_i K_i^T/\sqrt{d})V_i, \tag{2}$$

$$MultiHead(Q,K,V) = Concat(head_1,\dots,head_h)W^O, \tag{3}$$

where $W_q, W_k, W_v, W^O \in \mathbb{R}^{C\times C}$ are the projection matrices. $Q_i, K_i, V_i \in \mathbb{R}^{N\times d}$ are the query, key and value matrices in the i-th head, $d = C/h$, and h is the number of heads.

Glyph Multi-head Self-attention. The quadratic computational and memory complexity caused by self-attention is the primary problem for the generation of glyph images. We propose the glyph self-attention module to mitigate this problem, which attends to a small number of representative tokens by doing spatial sampling in the glyph feature map. As shown in Fig. 3, before computing multi-head self-attention, we sample $x \in \mathbb{R}^{N\times C}$ for h times ($N = H\times W$), each time sampling n tokens, where h is the number of heads and $n \ll N$. Denote Φ_i as the sampling function (i.e. torch.grid_sample function), it takes feature map $x \in \mathbb{R}^{N\times C}$ ($N = H\times W$) and a set of sampling coordinates (x axis, y axis) as inputs. $s_i \in \mathbb{R}^{n\times C}$ is the result sampled from $x \in \mathbb{R}^{N\times C}$, and i indexes the sampling times. The formula is as follows,

$$s_i = \Phi_i(x, coordinates),\ (1 \le i \le h) \tag{4}$$

For each s_i, we use different projection matrices to obtain the key and value matrices of h pair.

$$K_i = s_i W_k^i,\ V_i = s_i W_v^i, \tag{5}$$

where $W_k^i, W_v^i \in \mathbb{R}^{C\times d}$ are the projection matrices, K_i and V_i is the key and value matrices in the i-th head. The K_i (V_i) on different heads come from different sampling, please refer to Fig. 3(b) for details. The operation for query matrix is the same as the standard multi-head self-attention mentioned earlier.

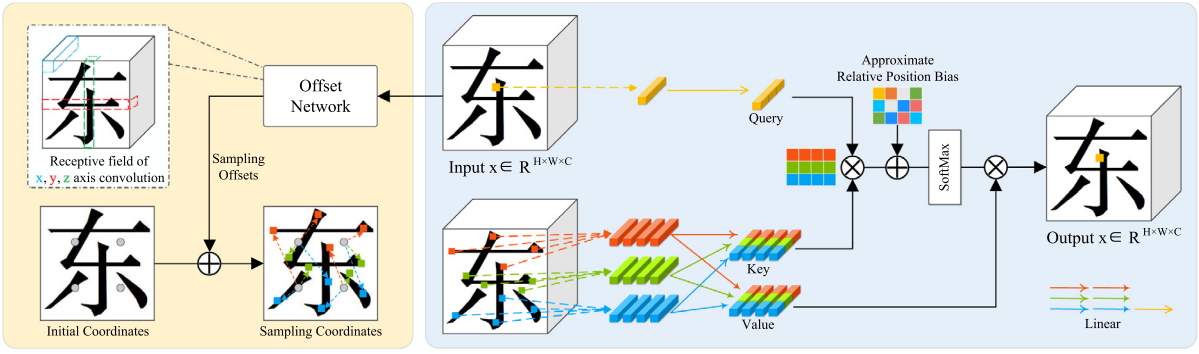
$$Q = xW_q,\ K = Concat(K_1,\dots,K_i),\ V = Concat(V_1,\dots,V_i), \tag{6}$$

Finally, the formula for the glyph multi-head self-attention is as follows,

$$head_i = SoftMax(Q_i K_i^T/\sqrt{d} + B)V_i, \tag{7}$$

$$MultiHead(Q,K,V) = Concat(head_1,\dots,head_h)W^O, \tag{8}$$

where B is the approximate relative position bias, which we describe in section 3.3. $W^O \in \mathbb{R}^{C\times C}$ is the projection matrices. Note that the



(a) Sampling by coordinates (b) Illustration of the glyph self-attention with three attention heads

Fig. 3. (a) The process of sampling by coordinates. The schematic diagram of the y, z axis convolution is shown in the upper left corner. It has a larger receptive field for sparse glyph feature maps but requires no extra parameters. (b) The illustration of the glyph self-attention with three attention heads, taking the update of a token in the input feature $x \in \mathbb{R}^{H\times W\times C}$ as an example. Multi-head multiple sampling enables different attention heads to attend to different locations explicitly.

symbols are the same as those in the standard multi-head self-attention formula mentioned earlier, unless otherwise noted.

Sampling by Coordinates. To make the model focus on the glyph area, it is critical to obtain the coordinates in the glyph areas. A naive approach is to use a lightweight convolutional network to predict coordinates directly, and we found that which is difficult to train in experiments. Inspired by anchor-based object detection methods (Redmon and Farhadi, 2017; Ren et al., 2015) and existing sampling methods (Xia et al., 2022; Yue et al., 2021; Zhu et al., 2020), we obtain the sampling coordinates by predicting the offset. Fig. 3(a) shows the process of sampling by coordinates, we first initialize n coordinates and then predict the corresponding offsets through a lightweight convolutional network, and the offset is added to the initial coordinates to obtain the sampling coordinates. The formula is as follows,

$$\Delta p = \theta_{offset}(x) \tag{9}$$

$$s_i = \Phi_i(x, p + \Delta p),\ (1 \le i \le h) \tag{10}$$

Eq. (11) is equal to Eq. (5), where $x \in \mathbb{R}^{H\times W\times C}$ is the input feature map, θ_{offset} is the lightweight convolutional network, $\Delta p \in \mathbb{R}^{h\times w\times 2}$ is the offset, and $p \in \mathbb{R}^{h\times w\times 2}$ is initial coordinates. p is evenly initialized as a rectangular grid with n coordinates ($n = h \times w$). The third dimension of p is 2, representing the x axis and y axis coordinates, respectively. Schematic diagram as shown in Fig. 3(a).

Convolution along the y, z Axis. There is another problem here. As shown in Fig. 3(a), the white area accounts for most of the glyph feature map. A reasonable assumption is that the accurate offset can only be output when the convolution kernel scans the glyph area, since conventional convolution struggles to capture glyph information when scanning non-glyph areas. Refer to the blue convolution kernel in Fig. 3(a) for intuitive understanding. Note that the offset network is a lightweight network. We want to minimize the number of its parameters. Therefore the receptive field cannot be increased by the number of layers. To address the above problem, we propose to perform convolution along the other two axes of the feature map, refer to the upper left of Fig. 3(a). The y, z axis convolution has a larger receptive field for sparse glyph feature maps but requires no extra parameters. For simplicity, we denote the conventional convolution as x axis convolution, and the convolution along the other two axes as y, z axis convolutions. The formula is as follows,

$$\Delta x = Conv\_x(x),\ \Delta y = Conv\_y(y),\ \Delta z = Conv\_z(z), \tag{11}$$

$$where\ x \in \mathbb{R}^{H\times W\times C} = y \in \mathbb{R}^{H\times C\times W} = z \in \mathbb{R}^{C\times W\times H} \tag{12}$$

$$\Delta p = Concat(\Delta y, \Delta z), \tag{13}$$

where $\Delta x \in \mathbb{R}^{h\times w\times 2}$, $\Delta y \in \mathbb{R}^{h\times w\times 1}$, $\Delta z \in \mathbb{R}^{h\times w\times 1}$. In practice, $Conv\_y$ and $Conv\_z$ is a 3×3 convolution followed by a 1×1 convolution, and their output channel is 1. To make full use of the information in the two axes, the outputs of $Conv\_y$ and $Conv\_z$ are concatenated as the offset.

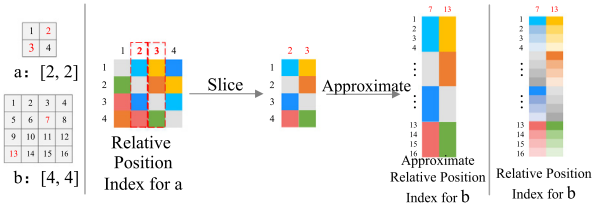


Fig. 4. Illustration of using the relative position bias of a to approximate the relative position bias of b, where a is the 2×2 feature map containing 4 tokens, and b is the 4×4 feature map containing 16 tokens. Assume that the sampled token is 7-th and 13-th in b, and the approximate relative position bias is using 2-th and 3-th in a to approximate 7-th and 13-th in b. Please compare the relative position index (Liu et al., 2021) of b shown on the right for a more intuitive understanding.

3.3. Approximate relative position bias

Our experiments show that the relative position bias (Liu et al., 2021) makes the model generate sharper glyph images. Since the relative position along each axis lies in the range [−H+1, W−1], a smaller-sized bias matrix $\hat{B} \in \mathbb{R}^{(2H-1)\times(2W-1)}$ is first parameterized, and then values in relative position bias B are taken from $\hat{B}$. However, the construction of the relative position index matrix suffers from the quadratic computational and memory complexity in the high-resolution stage. For a feature map of resolution $r \times r$, its relative position index matrix is $[r^2, r^2]$. On the other hand, unlike standard self-attention, which statically attends to all tokens, the glyph self-attention dynamically attends to the token at different locations. To address the above problem, we propose to approximate the high-resolution index matrix with the low-resolution index matrix. In practice, we use the relative position index matrix of 32×32 resolution to approximate the index matrix of 64×64 and 128×128 resolution.

For intuitive understanding, an example of using the index matrix of a:[2 × 2] to approximate the index matrix of b:[4 × 4] is shown in Fig. 4. Assuming that the target feature map is b, the 7-th and 13-th tokens are selected by the sampling function Φ to calculate the glyph self-attention. We perform the same sampling function Φ on a to obtain the 2-th and 3-th tokens in a, then initialize the relative position index matrix of a, and slice out the index matrix corresponding to the attention map of glyph self-attention. Finally, we expand the slice index

Fig. 5. Comparison of glyphs in five fonts generated by our method and the other four methods using eight reference samples on FZ-470.

matrix from a to obtain the approximate index matrix for b. Then we can use the approximate index to get the bias B from the bias matrix $\hat{B} \in \mathbb{R}^{(2H-1)\times(2W-1)}$ (Liu et al., 2021) and add it to the calculation of the glyph self-attention, refer to Eq. (8).

4. Experiments

4.1. Experiment setup

Datasets. Since there are no publicly available datasets, we collected a font dataset (**FZ-470**) with three categories of fonts (printed fonts, handwritten fonts, and calligraphy), a total of 470 fonts from FounderType,[1] each with 6,763 Chinese characters (Unicode 4E00~9FA5). All images are 256 × 256 in size. The Song San (a kind of font) is used as the source font, and other fonts are also available. We randomly select 400 fonts for pre-training and the remaining 69 fonts as new fonts for testing.

We further evaluate our model with another Chinese font library, SinoType.[2] An additional font dataset (**HW-18**) is collected for testing only, a total of 18 fonts, each with 6763 Chinese characters (Unicode 4E00~9FA5). All images are 256 × 256 in size.

Few-shot Setting. Take 8-shot as an example, we randomly select eight character images as the reference sample of each new font, and the test set of each new font consists of the remaining 6,755 characters. All subsequent experiments are conducted under the setting of 8-shot by default.

Evaluation Metrics. Pixel-level evaluation metrics are different from human perception and do not evaluate the generated glyph images well. In addition to RMSE, SSIM, and L1loss, we also employ perceptual-level metrics to evaluate the generated results. We train two ResNet-50 (He et al., 2016) classifiers on the test set of FZ-470 and HW-18 to recognize content and style, respectively. As the perceptual-level evaluation metrics, Err(C) and Err(S) represent content error and style error, respectively. The error is computed by the L2 distance of the feature vector between the generated glyph and ground truth. In addition, the perceptual-level metrics also include commonly used Frechet Inception Distance (FID) (Heusel et al., 2017). The prefix m of evaluation metrics represents the mean of the fonts in the test set.

Implementation Details. The number of transformer encoder layers is set to $N = 3$ in each transformer block, and each transformer encoder layer with four attention heads. All patch_size is set to 1. We use the nearest sampling for sampling function Φ in Eq. (5). We use the bicubic interpolation for the downsampling and upsampling layers. We use the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.95$ for pre-training and fine-tuning, and all learning rates are set to 0.0002. We use four NVIDIA Tesla V100 GPUs to pre-train our model. The pre-training takes about seven days, and fine-tuning takes only a few minutes.

4.2. Comparison with existing few-shot methods

Comparison Methods. We compare our method with the following state-of-the-art CNN-based methods for few-shot font generation, EMD (Zhang et al., 2018), LF-Font (Park et al., 2020), MX-Font (Park et al., 2021), and DG-Font (Xie et al., 2021). Note that LF-Font and MX-Font learn the glyph representation with the supervision of character radical annotations, while EMD, DG-Font, and our method do not require additional annotations. Except DG-Font can only generate glyph images with 80 × 80 resolution, other methods generate glyph images with 128 × 128 resolution.

Fairness. Although EMD, LF-Font, MX-Font, and DG-Font all claim to generalize to new fonts without fine-tuning, we fine-tune EMD and DG-Font to generate new fonts for fairness. Since our dataset does not support radical annotation, it is impossible to train LF-Font and MX-Font with FZ-470. We directly use their pre-trained models to generate new fonts. Their models are pre-trained with 467 fonts covering 19,234 Chinese characters, which is larger than the pre-training set of FZ-470 used in our method. We note here that LF-Font and MX-Font are not trained using our training set, which may have contributed to their weak performance. EMD and DG-Font are pre-trained with FZ-470, and the fine-tuned results are denoted as EMD* and DG-Font*.

Qualitative and Quantitative Evaluation. The visual comparisons on the two datasets are shown in Fig. 5 and Fig. 6. The results of EMD* are blurry. MX-Font and LF-Font used the same pre-training set, and while both are unsatisfactory, the former significantly outperforms the latter. The structure and style of the glyphs generated by DG-Font are inaccurate. After fine-tuning, DG-Font* is improved but still unsatisfactory. In contrast, the glyphs generated by our method have more accurate font style and stroke details, some glyphs and stroke details are marked by blue boxes and circles. The quantitative results are shown in Table 1. Both qualitative and quantitative comparisons show that our proposed TransFont significantly outperforms existing CNN-based models. Our method generate glyph images with more accurate font style and stroke details.

4.3. Ablation study

As mentioned in section 3.2 and 3.3, the proposed glyph self-attention differs from existing methods (Dai et al., 2017; Xia et al., 2022; Yue et al., 2021; Zhu et al., 2020) in three points, i.e., multi-head multiple sampling, yz axis convolution, and approximate relative position bias. To further evaluate the effectiveness of these techniques,

[1] http://www.foundertype.com/index.php/FindFont/index
[2] https://sinotype.vcg.com/

Fig. 6. Comparison of glyphs in five fonts generated by our method and the other four methods using eight reference samples on HW-18.

Table 1
Quantitative comparison of 8-shot font generation on FZ-470 and HW-18.

Dataset	Method	*m*Err(C)↓	*m*Err(S)↓	*m*FID↓	*m*RMSE↓	*m*SSIM↑	*m*L1loss↓
FZ-470	EMD*	0.3666	1.0299	177.14	0.6559	0.6127	0.2797
	LF-Font	0.4126	1.4847	198.61	0.8792	0.4962	0.4397
	MX-Font	0.2761	0.8264	236.47	0.8185	0.5265	0.3894
	DG-Font	0.2417	0.7836	156.23	0.6985	0.5813	0.2903
	DG-Font*	0.2279	0.7624	153.79	0.6583	0.6061	0.2893
	Ours	**0.1657**	**0.6738**	**142.17**	**0.6516**	**0.6146**	**0.2785**
HW-18	EMD*	0.1543	0.5446	174.05	0.4879	0.7345	0.1673
	LF-Font	0.2896	0.8431	180.14	0.7664	0.5748	0.3415
	MX-Font	0.1898	0.4609	199.38	0.6612	0.6319	0.2673
	DG-Font	0.2327	0.3997	126.51	0.4898	0.7053	0.1682
	DG-Font*	0.3651	0.3724	124.89	0.4791	0.7304	0.1688
	Ours	**0.0729**	**0.3404**	**115.53**	**0.4647**	**0.7441**	**0.1551**

Table 2
Quantitative comparison of the ablation study on FZ-470 and HW-18. *m.s.* represents the multi-head multiple sampling, *y.z.* represents the y, z axis convolution, and *a.rpb.* represents the approximate relative position bias.

Dataset	Setting	*m*Err(C)↓	*m*Err(S)↓	*m*FID↓	*m*RMSE↓	*m*SSIM↑	*m*L1loss↓
FZ-470	w/o *m.s.*	0.1786	0.9873	152.13	0.6607	0.6089	0.2833
	w/o *y.z.*	0.1871	0.9987	155.28	0.6613	0.6073	0.2846
	w/o *a.rpb.*	0.1818	0.9551	153.73	0.6595	0.6095	0.2803
	full model	**0.1657**	**0.6738**	**142.17**	**0.6516**	**0.6146**	**0.2785**
HW-18	w/o *m.s.*	0.0802	0.3994	123.89	0.4751	0.7368	0.1651
	w/o *y.z.*	0.0821	0.4056	125.37	0.4784	0.7359	0.1628
	w/o *a.rpb.*	0.0912	0.4524	126.82	0.4724	0.7383	0.1699
	full model	**0.0729**	**0.3404**	**115.53**	**0.4647**	**0.7441**	**0.1551**

we conduct the ablation study by separately removing these techniques from the glyph self-attention. The quantitative results of ablation experiments on FZ-470 and HW-18 are shown in Table 2.

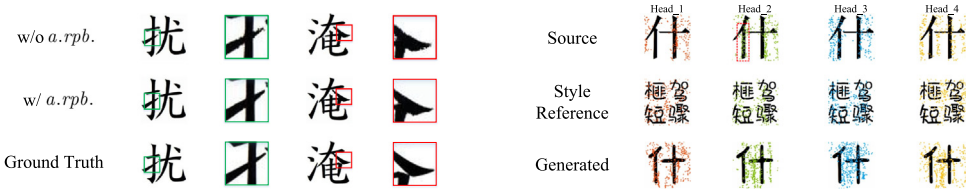
For the multi-head multiple sampling, we replace multiple sampling with single sampling and control other settings unchanged. The quantitative results have dropped on FZ-470 and HW-18. In addition to more sampling times, multiple sampling enables different attention heads to learn different representations explicitly, which is missing from multi-head attention with single sampling.

For the y, z axis convolution, we replace the y, z axis convolution with x axis convolution and control other settings unchanged. It can be seen that the quantitative results drop significantly on FZ-470 and HW-18. The unique sparsity of glyph images makes it difficult for x axis convolution to capture glyph information. This experiment shows that the y, z axis convolution mitigates this problem.

For the approximate relative position bias, we remove the relative position bias and control other settings unchanged. The quantitative results drop significantly on FZ-470 and HW-18. A more intuitive comparison is shown in Fig. 7(a), and the strokes generated by the model with *a.rpb.* are smoother, while the strokes generated by the model without *a.rpb.* are rough. This experiment shows the effectiveness of the proposed approximate relative position bias.

4.4. Interpretability of the glyph self-attention

The visualization of sampling locations is shown in Fig. 7(b). An obvious example on Head_2 in the source column is framed by a dashed red line, with the sample points concentrated in the glyph area. Here we draw three main conclusions. First, the glyph self-attention dynamically adjusts the sampling locations for different glyphs. Second, the sampling locations of different attention heads are different, which shows the effectiveness of the multi-head multiple sampling. Third, although white areas do not convey glyph information, the model still pays attention, indicating the importance of sparse representation of glyph images. The visualization results show that the interpretability of the proposed glyph self-attention. It is important to point out that the sampling location predicted by the offset network has a certain gap



(a) Effect of approximate relative position bias (b) Visualization of sampling locations

Fig. 7. (a) Comparison of glyphs generated by our model with or without *a.rpb.* (approximate relative position bias), the zoomed-in results are shown on the right. (b) Visualization of sampling locations in the glyph self-attention at the first transformer encoder layer of the content and style encoder, and the last transformer encoder layer of the decoder.

Table 3
Quantitative comparison with existing efficient self-attention modules on FZ-470 and HW-18.

Dataset	Attention	*m*Err(C)↓	*m*Err(S)↓	*m*FID↓	*m*RMSE↓	*m*SSIM↑	*m*L1loss↓
FZ-470	grid	0.1695	0.6818	145.16	0.6577	0.6131	0.2802
	multi-axis	0.1667	0.7811	149.45	0.6524	0.6144	0.2786
	deformable1	0.1669	0.8194	155.63	0.6579	0.6112	0.2809
	deformable2	0.1834	0.6758	144.57	0.6597	0.6113	0.2833
	glyph	**0.1657**	**0.6738**	**142.17**	**0.6516**	**0.6146**	**0.2785**
HW-18	grid	0.0789	0.3539	125.89	0.4721	0.7352	0.1615
	multi-axis	0.0788	0.3571	124.35	0.4728	0.7386	0.1621
	deformable1	0.0816	0.3555	123.14	0.4722	0.7385	0.1618
	deformable2	0.0828	0.3443	121.64	0.4725	0.7388	0.1637
	glyph	**0.0729**	**0.3404**	**115.53**	**0.4647**	**0.7441**	**0.1551**

Table 4
Inference latency (standard self-attention vs. glyph self-attention). Quantitative results are in milliseconds.

Resolution	64 × 64	128 × 128	256 × 256
Standard self-attention	6.590283121	111.249493315	**Out of Memory**
Glyph self-attention	6.679248226	9.452779509	22.343475214

from the ideal. Thus, some sampling locations is not concentrated in the glyph area.

4.5. Comparison with existing self-attention modules

Many methods have been proposed to mitigate the quadratic computational and memory complexity caused by the self-attention mechanism. We conducted a set of experiments, comparing the proposed glyph self-attention with existing four efficient self-attention modules (Jiang et al., 2021; Xia et al., 2022; Zhao et al., 2021; Zhu et al., 2020), to further evaluate the effectiveness of our method for glyph image generation. These self-attention modules include the grid self-attention proposed by TransGAN (Jiang et al., 2021) for image generation, the multi-axis self-attention proposed by HiT (Zhao et al., 2021) for image generation, the deformable attention proposed by DAT (Xia et al., 2022) for image classification, and the deformable attention proposed by deformable DETR (Zhu et al., 2020) for object detection. To distinguish the deformable attention of DAT and deformable DETR, we abbreviate them deformable1 (Xia et al., 2022) and deformable2 (Zhu et al., 2020), as shown in Table 3.

In experiments, we replace the glyph self-attention with the other four self-attention modules separately and control other settings to be the same. The quantitative results are shown in Table 3, and we can see that the glyph self-attention comprehensively outperforms the other four self-attention modules on FZ-470 and HW-18. Specifically, we found that the grid self-attention divides glyph images by grids, resulting in the truncation of generated glyphs. The edge of glyphs generated by multi-axis and deformable1 are not as sharp as ours, reflected in the metrics Err(S) and FID results. The character structure of glyphs generated by deformable2 is not as accurate as ours, reflected in the metrics Err(C) results. This experiment shows the effectiveness and robustness of our proposed glyph self-attention on glyph image generation.

4.6. Inference latency (standard self-attention vs. glyph self-attention)

Image generation employs pixel-level tokens for higher generation quality. In the high-resolution stage, the self-attention suffers from the quadratic computational and memory complexity. The glyph self-attention is proposed to mitigate this problem. To show the effectiveness of our method, we report inference latency following the code.[3]

The experimental setup is as follows, GPU: Tesla V100, 32G memory; embedding dim=128; attention_head=4; transformer encoder layer = 1. Quantitative results are in milliseconds. We take the input tensor at different resolutions, and test the inference latency of the self-attention module at different resolutions.

From Table 4, we can see that the inference latency of glyph self-attention is close to that of standard self-attention at 64 × 64 resolution. At 128 × 128 resolution, the inference latency of the proposed glyph self-attention is significantly faster. When the resolution is increased from 64 × 64 to 128 × 128, the inference latency of standard self-attention increased by a factor of 15.8, while our glyph self-attention only increased by a factor of 0.41. At 256 × 256 resolution, the inference latency of our glyph self-attention is acceptable, while standard self-attention is out of memory.

4.7. Effect of the selection of reference glyphs

The selection of reference samples has an impact on the generated results. We conducted a set of experiments on reference sample selection. We hand-picked three sets of reference samples (i.e., simple, middle, complex), each with eight characters. The characters in "simple" are extremely simple in structure and contain very few strokes. The characters in "middle" are moderately structured and contain a moderate number of strokes, similar to 8 random samples. The character structure in "complex" is extremely complex and contains a large number of strokes.

Table 6 report the number of stroke type and the total number of strokes in each group. Table 5 shows the quantitative results of the

[3] https://deci.ai/blog/measure-inference-time-deep-neural-networks

Fig. 8. Comparison of glyphs in nine fonts generated by our method in different few-shot setting.

Table 5
Quantitative comparison of the selection of reference glyphs on FZ-470 and HW-18.

Dataset	Patch_size	*m*Err(C)↓	*m*Err(S)↓	*m*FID↓	*m*RMSE↓	*m*SSIM↑	*m*L1loss↓
FZ-470	Simple	1.5901	1.1543	169.27	0.7824	0.5384	0.3763
	Middle	0.1735	0.6828	143.61	0.6552	**0.6148**	0.2741
	Complex	0.1806	0.6759	144.97	0.6524	0.6137	**0.2739**
	Random	**0.1657**	**0.6738**	**142.17**	**0.6516**	0.6146	0.2785
HW-18	Simple	0.0869	0.4909	121.72	0.4984	0.7188	0.1772
	Middle	0.0739	0.3573	114.76	0.4649	0.7403	0.1568
	Complex	**0.0714**	**0.3481**	**112.39**	**0.4622**	0.7426	**0.1544**
	Random	0.0729	0.3495	115.53	0.4647	**0.7441**	0.1551

Table 6
Four group of reference glyphs, this table shows the number of stroke type and the total number of strokes in each group.

Group	Number of stroke type	Total number of strokes
Simple	12	32
Middle	14	89
Complex	18	160
Random	15	83

selection of reference glyphs on FZ-470 and HW-18. We can draw the following conclusions,

(1) When the structure of the reference sample is extremely simple, or the selected sample contains a very small number of strokes, the generation quality is poor;

(2) The generated results of "middle" and "complex" are close, and the results of "complex" are relatively better but limited.

(3) It shows that the more strokes the reference sample contains, the better the result. This gap may be indistinguishable by the human eye from "middle" to "complex".

(4) The generated results of "random" are acceptable. In other words, the strategy of random reference samples is a good measure for model performance. This experiment shows the rationality of our experiment setup for new fonts.

4.8. Effect of the number of reference samples

The number of new font reference samples also affects the generation results. We conducted a set of experiments to test the effect of the number of reference samples on the generation quality. Fig. 9 shows the visual generated results under different few-shot settings. The visual results show that our model can handle various new font generation in different few-shot settings. As the number of reference samples increases, some strokes are generated better. Fig. 8 presents quantitative results on two datasets, FZ-470 and HW-18. The quantitative results change significantly when the number of reference samples is very small (1, 4, 8). The quantitative results tend to be stable when the number of samples is relatively large (16, 32, 64, and 128). This experiment shows that our transformer-based model is good at few-shot learning. The performance of our method is stable, and it is robust to various new font generation.

4.9. Failure case

The complexity of the new font is the key to restricting the model performance, and our model performs poorly on some complicated fonts. Fig. 10 shows the failure case on both fonts. We think the poor performance is mainly due to the high irregularity of the font style. Generating such a kind of cursive calligraphy font may require some domain knowledge of calligraphy.

5. Conclusion

In this work, based on the sparsity of glyph images, we introduced the glyph self-attention module for efficiently representing glyph images in the self-attention mechanism, offering insights for future transformer-based font generation models. On top of this, we presented the TransFont, a simple but effective transformer baseline for few-shot font generation, showing its superiority over existing CNN-based font generation models on two challenging Chinese font libraries, FounderType and SinoType. One limitation of our method is that it relies on fine-tuning to generate new fonts. In some application scenarios, fine-tuning is not supported. Generating high-quality new fonts without fine-tuning will be future work.

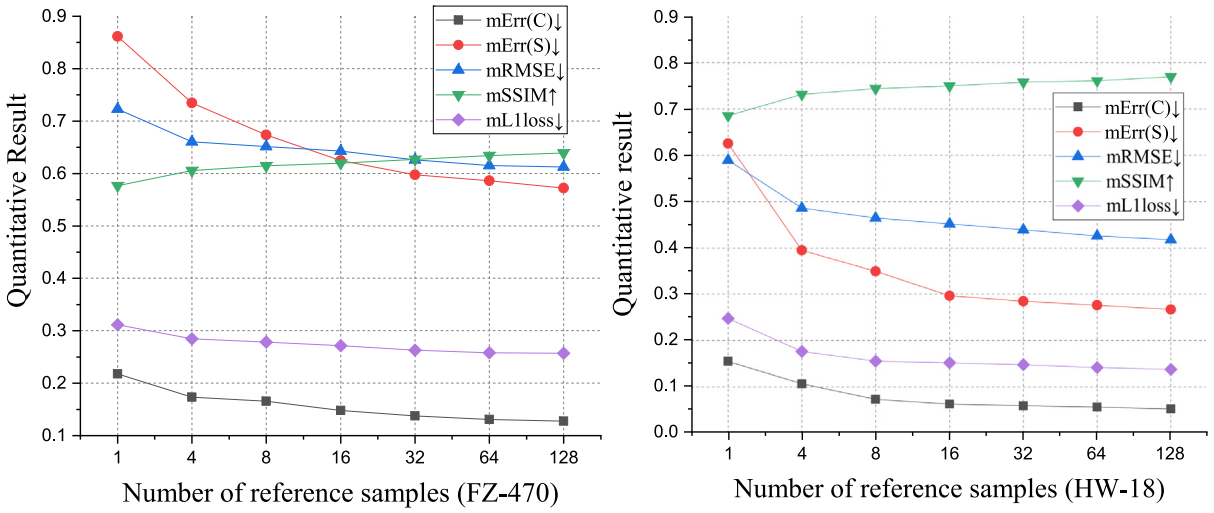


Fig. 9. Quantitative results about the effect of the number of reference samples. With the increase of the number of reference samples, the generation quality gradually improves and then tends to be stable.

Fig. 10. Generated results on two complicated Chinese fonts, the left font is from the ancient Chinese litterateur "Su Shi", and the right is an irregular font.

CRediT authorship contribution statement

Xu Chen: Methodology, Writing – original draft. **Lei Wu:** Supervision, Writing – review & editing. **Yongliang Su:** Validation. **Lei Meng:** Funding acquisition. **Xiangxu Meng:** Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported in part by the National Key R&D Program of China (Grant no. 2021YFC3300203) and the Oversea Innovation Team Project of the "20 Regulations for New Universities" funding program of Jinan (Grant no. 2021GXRC073).

References

Azadi, S., Fisher, M., Kim, V.G., Wang, Z., Shechtman, E., Darrell, T., 2018. Multi-content gan for few-shot font style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7564–7573.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language models are few-shot learners 33. pp. 1877–1901.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. In: European Conference on Computer Vision. Springer, pp. 213–229.

Cha, J., Chun, S., Lee, G., Lee, B., Kim, S., Lee, H., 2020. Few-shot compositional font generation with dual memory. In: European Conference on Computer Vision. Springer, pp. 735–751.

Chang, B., Zhang, Q., Pan, S., Meng, L., 2018. Generating handwritten chinese characters using cyclegan. In: 2018 IEEE Winter Conference on Applications of Computer Vision. WACV, IEEE, pp. 199–207.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I., 2020. Generative pretraining from pixels. In: International Conference on Machine Learning. PMLR, pp. 1691–1703.

Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W., 2021a. Pre-trained image processing transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12299–12310.

Chen, X., Wu, L., He, M., Meng, L., Meng, X., 2021b. MLFont: Few-shot Chinese font generation via deep meta-learning. In: Proceedings of the 2021 International Conference on Multimedia Retrieval. pp. 37–45.

Child, R., Gray, S., Radford, A., Sutskever, I., 2019. Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y., 2017. Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 764–773.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B., 2021. Cswin transformer: A general vision transformer backbone with cross-shaped windows. arXiv preprint arXiv:2107.00652.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16 × 16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Esser, P., Rombach, R., Ommer, B., 2021. Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12873–12883.

Gao, Y., Guo, Y., Lian, Z., Tang, Y., Xiao, J., 2019. Artistic glyph image synthesis via one-stage few-shot learning. ACM Trans. Graph. 38 (6), 1–12.

Gao, Y., Wu, J., 2020. Gan-based unpaired chinese character image translation via skeleton transformation and stroke rendering. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, No. 01. pp. 646–653.

Gatys, L.A., Ecker, A.S., Bethge, M., 2016. Image style transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2414–2423.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W., 2018. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. Adv. Neural Inf. Process. Syst. 27.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S., 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Adv. Neural Inf. Process. Syst. 30.

Huang, Y., He, M., Jin, L., Wang, Y., 2020. RD-GAN: few/zero-shot Chinese character style transfer via radical decomposition and rendering. In: European Conference on Computer Vision. Springer, pp. 156–172.

Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1125–1134.

Jiang, Y., Chang, S., Wang, Z., 2021. Transgan: Two pure transformers can make one strong gan, and that can scale up. Adv. Neural Inf. Process. Syst. 34.

Jiang, Y., Lian, Z., Tang, Y., Xiao, J., 2017. DCFont: an end-to-end deep Chinese font generation system. In: SIGGRAPH Asia 2017 Technical Briefs. pp. 1–4.

Jiang, Y., Lian, Z., Tang, Y., Xiao, J., 2019. Scfont: Structure-guided chinese font generation via deep stacked networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, No. 01. pp. 4015–4022.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kong, Y., Luo, C., Ma, W., Zhu, Q., Zhu, S., Yuan, N., Jin, L., 2022. 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. CVPR, pp. 13482–13491.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022.

Liu, W., Liu, F., Din, F., He, Q., Yi, Z., 2022. XMP-font: Self-supervised cross-modality pre-training for few-shot font generation. arXiv preprint arXiv:2204.05084.

Lyu, P., Bai, X., Yao, C., Zhu, Z., Huang, T., Liu, W., 2017. Auto-encoder guided GAN for Chinese calligraphy synthesis. In: 2017 14th IAPR International Conference on Document Analysis and Recognition, Vol. 1. ICDAR, IEEE, pp. 1095–1100.

Naseer, M.M., Ranasinghe, K., Khan, S.H., Hayat, M., Shahbaz Khan, F., Yang, M.H., 2021. Intriguing properties of vision transformers. Adv. Neural Inf. Process. Syst. 34.

Park, S., Chun, S., Cha, J., Lee, B., Shim, H., 2020. Few-shot font generation with localized style representations and factorization. arXiv preprint arXiv:2009.11042.

Park, S., Chun, S., Cha, J., Lee, B., Shim, H., 2021. 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.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., 2018. Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al., 2019. Language models are unsupervised multitask learners. OpenAI Blog 1 (8), 9.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I., 2021. Zero-shot text-to-image generation. In: International Conference on Machine Learning. PMLR, pp. 8821–8831.

Redmon, J., Farhadi, A., 2017. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7263–7271.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. 28.

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z., 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1874–1883.

Sun, D., Ren, T., Li, C., Su, H., Zhu, J., 2017. Learning to write stylized chinese characters by reading a handful of examples. arXiv preprint arXiv:1712.06424.

Tang, L., Cai, Y., Liu, J., Hong, Z., Gong, M., Fan, M., Han, J., Liu, J., Ding, E., Wang, J., 2022. Few-shot font generation by learning fine-grained local styles. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 7895–7904.

Tuli, S., Dasgupta, I., Grant, E., Griffiths, T.L., 2021. Are convolutional neural networks or transformers more like human vision? arXiv preprint arXiv:2105.07197.

Vaswani, A., Ramachandran, P., Srinivas, A., Parmar, N., Hechtman, B., Shlens, J., 2021. Scaling local self-attention for parameter efficient visual backbones. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12894–12904.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Wen, Q., Li, S., Han, B., Yuan, Y., 2021. ZiGAN: Fine-grained Chinese calligraphy font generation via a few-shot style transfer approach. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 621–629.

Wu, L., Chen, X., Meng, L., Meng, X., 2020a. Multitask adversarial learning for Chinese font style transfer. In: 2020 International Joint Conference on Neural Networks. IJCNN, IEEE, pp. 1–8.

Wu, X., Hu, Z., Sheng, L., Xu, D., 2021. StyleFormer: Real-time arbitrary style transfer via parametric style composition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14618–14627.

Wu, S.J., Yang, C.Y., Hsu, J.Y.j., 2020b. Calligan: Style and structure-aware chinese calligraphy character generator. arXiv preprint arXiv:2005.12500.

Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G., 2022. Vision transformer with deformable attention. arXiv preprint arXiv:2201.00520.

Xie, Y., Chen, X., Sun, L., Lu, Y., 2021. DG-font: Deformable generative networks for unsupervised font generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5130–5140.

Yue, X., Sun, S., Kuang, Z., Wei, M., Torr, P.H., Zhang, W., Lin, D., 2021. Vision transformer with progressive sampling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 387–396.

Zeng, J., Chen, Q., Liu, Y., Wang, M., Yao, Y., 2021. Strokegan: Reducing mode collapse in chinese font generation via stroke encoding. In: Proceedings of AAAI, Vol. 3.

Zhang, B., Gu, S., Zhang, B., Bao, J., Chen, D., Wen, F., Wang, Y., Guo, B., 2021. StyleSwin: Transformer-based GAN for high-resolution image generation. arXiv preprint arXiv:2112.10762.

Zhang, Y., Zhang, Y., Cai, W., 2018. Separating style and content for generalized style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8447–8455.

Zhao, L., Zhang, Z., Chen, T., Metaxas, D., Zhang, H., 2021. Improved transformer for high-resolution gans. Adv. Neural Inf. Process. Syst. 34.

Zhu, J.Y., Park, T., Isola, P., Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2223–2232.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J., 2020. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159.