IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 28 March 2024, accepted 22 April 2024, date of publication 29 April 2024, date of current version 6 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3394441

RESEARCH ARTICLE

MA-Font: Few-Shot Font Generation by Multi-Adaptation Method

YANBO QIU[1], KAIBIN CHU[1], JI ZHANG[2], AND CHENGTAO FENG[1]
[1]School of Microelectronics and Control Engineering, Changzhou University, Changzhou 213159, China
[2]School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou 213159, China

Corresponding author: Kaibin Chu (ckb910@163.com)

Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant (KYCX23_3182).

ABSTRACT Few-shot font generation (FFG) stands as a pivotal technique in Chinese character generation, enabling the creation of new fonts by leveraging a limited set of available font images. Despite the remarkable success of existing cross-language font generation methods, they tend to ignore some domain-specific characteristics. In addition, they cannot achieve one-to-many language conversion, that is, they cannot give the same style to texts in different languages. Therefore, this paper introduces a novel end-to-end method for Chinese character generation that aims to achieve cross-language font generation. This method incorporates three pivotal modules: the content self-adaptation module, the multi-Head attention module, and the co-adaptation module. The content self-adaptation module preserves the semantic structure of the content image by capturing spatial similarities in arbitrary positions in the content feature map. The multi-head attention module is used to capture local and global features of the style reference images. Finally, the co-adaptation module reorganizes the captured style features based on the semantic structure of the content image to generate new features. In comparative experiments, our model demonstrates superior overall performance compared to existing cross-lingual font generation methods.

INDEX TERMS Font generation, self-attention, multi-adaptation, few-shot font generation, style transfer, image-to-image translation.

I. INTRODUCTION

Chinese characters, as the most widely used script in East Asia, serve as a carrier for the transmission of Chinese culture, possessing intricate character structures and semantics. When designing a novel typeface, the primary concern is ensuring that humans can accurately recognize the characters, followed by the pursuit of artistic appeal in the typeface design. Due to the complex structure and vast quantity of Chinese characters, constructing a commercial typeface library is expensive and labor-intensive. Designing a complete typeface library with only a few character forms is nearly impossible for someone without artistic knowledge.

In recent years, with the development of deep learning, pioneers have made significant progress in font generation using convolutional neural networks [1] and generative adversarial networks (GANs) [2], generating satisfactory fonts. Inspired by deep neural networks, Tian et al. proposed "Rewrite" [3], a model that utilizes the structure of a CNN to generate fonts similar to the target font. Subsequently, Zi2zi [4] was introduced, which incorporates font category embedding conditions based on pix2pix [5] to model one-to-many relationships. In [6], the author uses a component encoder to extract the structural information of Chinese characters and adds these component information to the model. In [7], the author designs a self-attentive refined attention module to extract the skeleton information of calligraphy. These models require large numbers of paired samples, but collection of paired samples is labor-intensive and expensive. Especially in some Chinese character generation tasks, such as the generation of Chinese calligraphy fonts and Mongolian fonts.

Some Chinese character generation techniques [8], [9], [10], [11] attempt to improve the results of Chinese character generation using image-to-image translation (I2I) methods. For example, HCCG-CycleGAN [12] and StrokeGAN [13] both utilize the CycleGAN [14] as the main framework for

The associate editor coordinating the review of this manuscript and approving it for publication was Zeev Zalevsky.

Chinese character generation tasks. The former employs an encoder-decoder structure for the generator and incorporates DenseNet [15] to capture the high-frequency information of fonts, while the latter introduces a one-hot stroke encoding to capture the key information of Chinese characters. In subsequent developments, the author of StrokeGAN introduces two notable extensions, namely SGCE-Font [16] and StrokeGAN+ [17], building upon the foundation laid by the original StrokeGAN framework. SGCE-Font [16] introduces the Skeleton Guided Channel Expansion (SGCE) module, a novel addition to the generator architecture. StrokeGAN+ [17] integrates a "few-sample semi-supervised scheme" to improve font generation performance by utilizing limited labeled data. However, these methods still suffer from missing stroke and redundancy issues.

Style and content representation are essential for few-shot font generation. Some methods [18], [19], [20], [21] focus on disentangling the representation of content and style. These methods can effectively transform the style of content images to match the style of characters in the source domain or to achieve the target style for known or unknown fonts. Typically, these methods employ two separate encoders to learn content and style features.

However, all described methods cannot transfer styles between different languages. In most cases, having multilingual fonts with the same style is very important, but also more challenging. For example, when signing a cross-border business contract, it is necessary to ensure that texts in different languages have the same style. Therefore, it is necessary to achieve cross-language style transfer. In addition, the differences in character structures between different languages bring great challenges to the cross-language font generation task. For example, the structure of some Chinese characters is very complex, while the structure of Latin letters is relatively simple. Recently, a work [22] attempted to resolve the character differences of different languages to achieve cross-language font generation. These works ignore some specific domain features, resulting in insensitivity to local details of some specific fonts, such as handwriting. In addition, these works cannot achieve one-to-many language conversion, that is, they cannot give the same style to texts in different languages.

Therefore, this paper proposes a novel end-to-end Chinese character style transfer model called MA-Font for cross-language font style transfer. This model preserves the structural information of characters in the source domain while transferring the style of the reference font to the generated results.

Fig.1 shows an example application of MA-Font. The proposed method generates a famous poem, "Prelude to Water Melody." On the left side are the style reference image and the character image, while on the right side are the generated result and the Ground Truth.

Our contributions can be summarized as follows:

1. This paper proposes a novel end-to-end Chinese character style transfer model called MA-Font for cross-language font style transfer. Specifically, this paper calculates the correlation between the content and style features of fonts, and then rearranges the style features according to the content feature distribution.

2. This paper introduces three novel modules: a content self-adaptation module, multi-head attention module, and co-adaptation module, to enhance the representation of content features and style features.

3. In this work, we conduct multiple experiments using a multi-language glyph image dataset containing 847 fonts. Additionally, we demonstrate the performance of the proposed model in Chinese character generation tasks through visual quality analysis and quantitative evaluation.

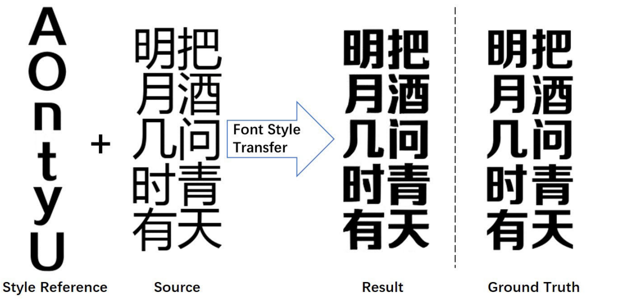

FIGURE 1. The proposed method generates a famous poem, "Prelude to Water Melody." On the left side are the style reference image and the character image, while on the right side are the generated result and the Ground Truth.

The structure of the remaining parts of this paper is as follows: Section II describes related work, providing a brief introduction to existing methods for image-to-image translation and few-shot font generation. Section III presents the proposed model in detail. Section IV provides a comprehensive overview of the experimental setup, results, and comparative data. The conclusion of this paper is presented in Section V.

II. RELATED WORKS

A. IMAGE-TO-IMAGE TRANSLATION

The task of image-to-image translation (I2I) is to translate the style of a source domain into the style of a reference image. Since the introduction of a method by Gatys et al. [24] that utilizes CNNs for style transfer, many efforts are being made to improve the efficiency and quality of image translation. For example, CycleGAN [14] uses mapping between two domains to achieve style transfer, but these methods require expensive data.

Later, some works achieve style transfer between images by separating content features from style features, and this approach gains widespread application. SC-GAN [25] proposes a new unsupervised algorithm to learn disentangled style and content representations of the data. One work [26] achieves artistic style transfer by separating style and content with two new loss functions. DMIT [27] decomposes the input image into latent representations and then achieves multi-domain translation by manipulating different parts of the latent representations.

FUNIT [28] proposes an unsupervised image-to-image translation framework with a few-shot samples, using two encoders to extract features from the content and style images. Inspired by DaNet [29], Deng et al. [30] design a flexible and efficient style transfer model that uses a novel disentanglement loss function to extract style and content information from images. Huang et al. [31] introduce adaptive instance normalization (AdaIN), which adjusts the mean and variance of the content to adapt to the style image. Kitov et al. [32] directly control the intensity of stylization by using a network of transformers.

Although there are many similarities between image-to-image translation and font style transfer, image-to-image translation cannot be directly applied to font style transfer. The former primarily focuses on the color and texture of style images, while the latter requires preserving the structural information of characters. The methods of image-to-image translation bring substantial inspiration to researchers. Through image-to-image translation, researchers can delve into the intrinsic feature representation and transfer methods of Chinese character glyphs, leading to the design of more effective methods for generating Chinese characters.

B. ATTENTION MECHANISM

In this article, the self-attention mechanism is the key technology of MF-Font, so some related work on the self-attention mechanism is selectively introduced in this section.

The attention mechanism [33], [34] is an important component of deep learning and is widely used in many tasks. It plays an important role in the field of natural language processing, allowing models to dynamically focus on different parts of input text to handle relationships and context between texts. Later, Yang et al. [35] apply the attention mechanism to text classification tasks, allowing the model to distinguish and focus on different text contents when building document representations. Yu et al. [36] design a multi-level attention network based on the attention mechanism, which can obtain semantic information from a single image and reduce the semantic gap through semantic attention.

Recently, the work [37] proposes the self-attention mechanism for the first time and achieves a breakthrough in the field of natural language processing. SAGAN [38] applies self-attention to generative adversarial networks for the first time, significantly improving the image generation task. At the same time, self-attention mechanisms are increasingly used in image generation tasks. SAnet [39] proposes a feed-forward network to match similar style features to content features. DAnet [29] uses a position attention module and a channel attention module to build a dual attention network. MCCNet [40] uses a self-attention mechanism to integrate content features and style features. Here, the proposed method uses the self-attention mechanism to build three modules.

C. FONT GENERATION

Chinese character generation is a challenging task that can be viewed as transferring the style of one font to another. In a sense, Chinese character style transfer is a special case of image-to-image translation. Some existing font generation methods are based on the framework of image-to-image translation, which can be broadly classified into many-shot font generation methods and few-shot generation methods.

Many-shot font generation methods require a large number of paired datasets, but creating such datasets involves high costs, as seen in approaches like Zi2zi [4], DC-Font [41] and SC-Font [42]. zi2zi is not published in the form of a paper, but it has a significant impact on font generation. DC-Font expands upon zi2zi by utilizing a VGG network for pretraining on a diverse set of 100 font types, effectively extracting distinctive style features from each. Ultimately, it combines the font style categories with the output of the encoder to achieve its desired results. SC-Font builds upon DC-Font by incorporating a specialized stroke extraction algorithm to extract stroke features, which are utilized to guide style transformation.

Existing few-shot font generation methods [43], [44], [45], [46], [47] take as input the content and style images, and generate results that resemble the target font. EMD [48] and AGISNet [19] simply combine style vectors and content vectors as input to the decoder for generating the target characters. MX-Font [49] utilizes multiple experts to extract style features. DG-Font [50] introduces variability blocks to improve the results of Chinese character generation. LF-Font [51] designs a style encoder with component conditions to capture style features. However, the described methods demonstrate the effectiveness of generating new characters using several style reference characters, but they can only perform style transfer between the same languages.

Both the proposed MA-Font and the aforementioned methods are GAN-based approaches. However, MA-Font has two notable distinctions. Firstly, MA-Font rearranges style features while preserving specific domain characteristics. Secondly, MA-Font can be applied to cross-language font generation.

III. METHOD

This section mainly introduces the method of MA-Font to generate multi-language fonts. In simple terms, the typical task of generating glyph images can be seen as mapping a given content image I_c and a set of style images I_s that maintain different styles but with the same content to the target font image. In this process, the content encoder E_c and the style encoder E_s extract corresponding feature maps $Z_c = E_c(I_c)$ and $Z_s = E_s(I_s)$, respectively. Then, the extracted feature information is fed into the decoder to generate the target glyph image.

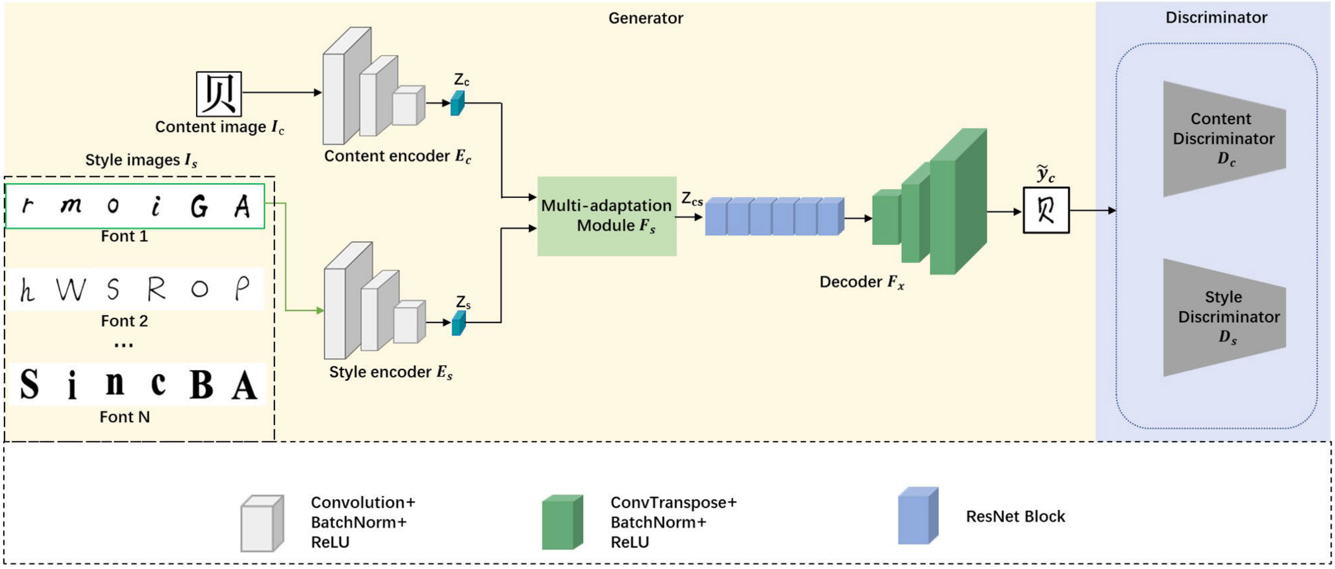

FIGURE 2. Overview of MA-Font. The style (content) encoder extracts corresponding feature maps from the style (content) image. Then, the multi-adaptation module rearranges the style features based on the distribution of content features. Finally, it generates high-quality stylized images.

This task can be represented by the following equation:

$$G(I_c, I_s) \rightarrow F_{gt} \tag{1}$$

where I_c represents the given content image, I_s represents the style images, and F_{gt} represents the Ground Truth.

However, considering the domain discrepancy between the content images and style images, using a generic encoder can only capture a limited amount of content and style features. To address this, this paper introduces a multi-adaptation module that can rearrange the style features according to the distribution of content features through an adaptive process. This enables us to obtain glyph images with stylized effects.

A. NETWORK OVERVIEW

Given a content image I_c, where the content C comes from a set of standard fonts $X = \{I_c\}_{c=1}^{N}$, and a set of style images $I_s = \{y_i\}_{i=1}^{k}$, our model aims to generate stylized images $\tilde{y}_c$ using the generator G, where $\tilde{y}_c$ should combine the content C and style S. Considering that it is difficult to extract a common style from a single style reference image, the proposed method requires randomly selecting 6 style reference images as style input.

The process can be defined as follows:

$$\tilde{y}_c = G(I_c, I_s) \tag{2}$$

The proposed Chinese character generation model, as shown in Fig.2, adopts a Generative Adversarial Network (GAN) framework that includes a generator and two discriminators. The two discriminators are the content discriminator D_c and the style discriminator D_s. The main task of the generator G is to generate target glyph images through extensive training. On the other hand, the discriminators D_c and D_s serve to differentiate between the generated font images and real images. The generator continuously optimizes the generated font images to deceive the discriminators, while the discriminators continuously optimize themselves to prevent the generator from producing fake images that can pass through them. This adversarial relationship between the generator and discriminators drives the model to improve the quality of generated images over time.

In order to better integrate intermediate and high-level semantic features, the generator adopts an encoder-decoder structure, which helps capture style images at different scales during context fusion. The generator consists of a content encoder E_c, a style encoder E_s, a multi-adaptation module F_s, and a decoder F_x. The content encoder E_c consists of three convolutional blocks, with kernel sizes 7, 3, and 3 respectively. BatchNorm and ReLU activation functions are applied after each convolutional block. The content encoder maps the input content image I_c to latent codes Z_c. Similarly, the style encoder E_s has a similar structure to the content encoder and it will extract the feature mapping Z_s for k style images. Subsequently, the latent codes Z_c and feature maps Z_s are input into the multi-adaptation module F_s, which reorganizes the style features based on the content feature distribution to generate new style features Z_{cs}. For more details, please refer to Section III-B.

The decoder F_x consists of six Resnet blocks and three transpose convolutional layers, similar to the encoder. BatchNorm and RelU activations follow each convolutional layer. Tanh activation is used as the final layer activation in the decoder.

The architecture and layer specifications of the encoders, decoder and discriminator can be observed in Table 1.

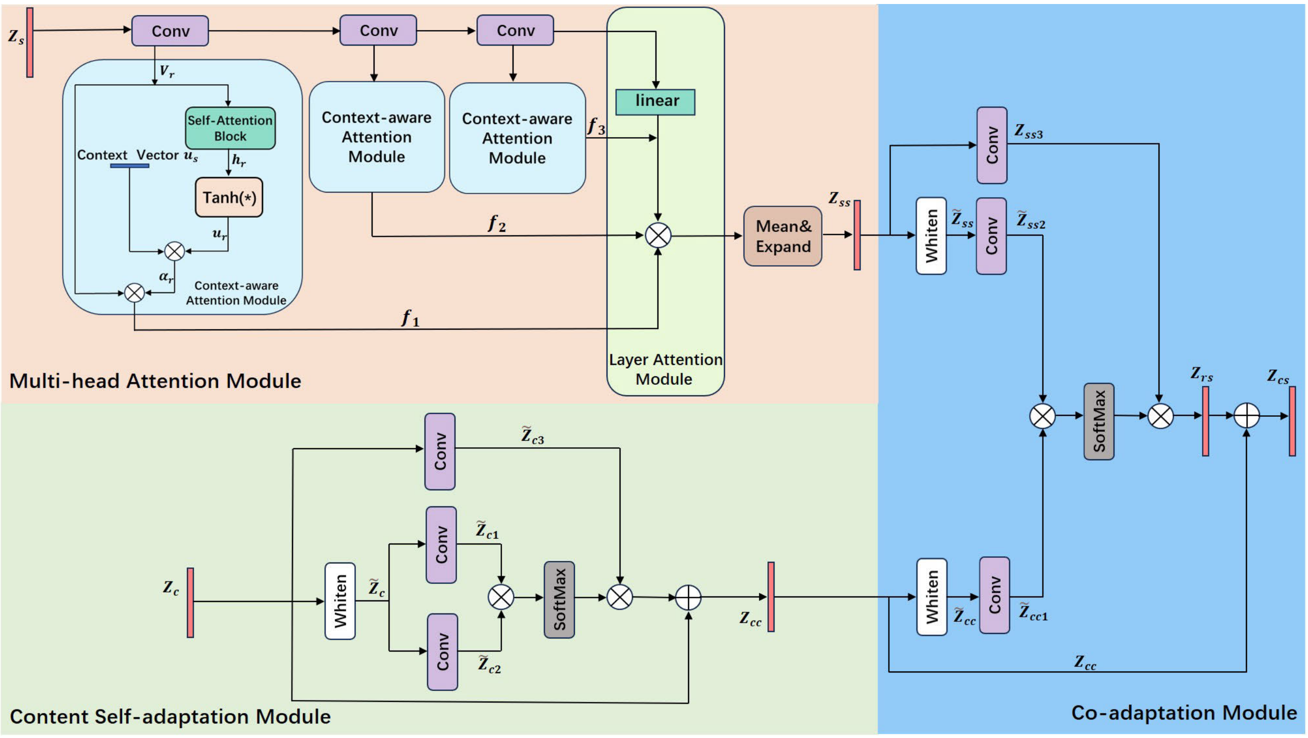

FIGURE 3. Multi-adaptation Module. The multi-adaptation module consists of the multi-head attention module, content self-adaptation module, and co-adaptation module. Additionally, we set the background of the multi-head attention module to red, the background of the content self-adaptation module to green, and the background of the co-adaptation module to blue.

TABLE 1. The layer specifications for encoders, decoder and discriminator.

Module	Specifications
Content Encoder	7*7 Conv-BN-Rule, 64 filters, stride 1 3*3 Conv-BN-Rule, 128 filters, stride 2 3*3 Conv-BN-Rule, 256 filters, stride 2
Style Encoder	7*7 Conv-BN-Rule, 64 filters, stride 1 3*3 Conv-BN-Rule, 128 filters, stride 2 3*3 Conv-BN-Rule, 256 filters, stride 2
Decoder	6 ResNet Block, 512 filters 3*3 Deconv-BN-Rule, 256 filters, stride 2 3*3 Deconv-BN-Rule, 128 filters, stride 2 7*7 Deconv-Tanh, 1 filters stride 1
Discriminator	4*4 Conv-Leaky Rule,64 filters, stride 2 4*4 Conv-BN-Leaky Rule, 128 filters, stride 2 4*4 Conv-BN-Leaky Rule, 256 filters, stride 2 4*4 Conv-BN-Leaky Rule, 512 filters, stride 1 4*4 Conv, 1 filters, stride 1

The following formulas express the operations of E_c, E_s, F_s and F_x:

$$\tilde{y}_c = F_x(F_s(Z_c, Z_s)) = F_x(F_s(E_c(I_c), E_s(I_s))) \qquad (3)$$

where $\tilde{y}_c$ represents the stylized images, Z_s and Z_c denote the outputs of the style encoder E_s and content encoder E_c, respectively.

B. MULTI-ADAPTATION MODULE

One of the key aspects in Chinese character generation is that the generated glyph images should contain the structural information of the content characters for proper font recognition, while also incorporating the stylistic information from the style images to fulfill the aesthetic requirements of human observers. Inspired by DaNet [29], we design a module that not only aggregates styles appropriately but also focuses on the stylistic expressiveness of the style reference image's local details. Instead of directly concatenating the features extracted by the encoder and feeding them to the decoder, this module enables effective integration of style while considering the nuanced and localized style expressions.

Fig.3 displays the details of the multi-adaptation module. The module primarily consists of three parts: the multi-head attention module in red, the content self-adaptation module in green, and the co-adaptation module in blue. The multi-adaptation module takes the feature maps from the last convolutional layer of the content encoder and style encoder as input. Through the multi-head attention module and content self-adaptation module, the respective style features Z_s and content features Z_c can be represented as Z_{ss} and Z_{cc}. Subsequently, the co-adaptation module recombines Z_{ss} and Z_{cc} to generate stylized feature Z_{cs}.

1) MULTI-HEAD ATTENTION MODULE

To effectively establish the contextual relationship between local features and global features in the font style task, we draw inspiration from FTransGAN [52] and MF-Net [53], and evolve it into an adaptive multi-level attention mechanism. The multi-head attention module consists of three context-aware attention modules and a layer attention module, which can map the style images $I_s = \{y_i\}_{i=1}^{k}$ to intermediate feature vectors. Subsequently, the final style feature vector Z_{ss} is obtained by calulating the average of these intermediate feature vectors.

The feature map Z_s of the last convolutional layer of the style encoder is used as the input to the context-aware module, enabling the preservation of contextual information for each region. Therefore, the feature map Z_s can be expressed as $\{V_r, r = 1, 2, \cdots, H \times W\}$, where r represents the feature vector of the r-th region, and H and W represent the height and width, respectively.

When encoding the region feature vector V_r, the self-attention layer takes into account the contextual relationships between adjacent regions and incorporates relevant contextual information into the new region feature vector h_r.

$$h_r = SA(V_r) \tag{4}$$

where SA represents the self-attention layer, h_r represents the new feature vector after merging contextual information. h_r includes the contextual information around the r-th region, but primarily focuses on the r-th region.

Furthermore, in order to reward the inclusion of accurate contextual information in each region, an attention mechanism and context vector u_s are used to measure the contribution of each region in terms of contextual information.

The feature vector h_r adjusts attention weights by optimizing the parameters of the single-layer MLP.

$$u_r = \tanh(w_s h_r + b_s) \tag{5}$$

where w_s and b_s is the parameter of the single-layer MLP.

$$\alpha_r = \frac{\exp(u_r^T u_s)}{\sum_{H \times W} \exp(u_r^T u_s)} \tag{6}$$

where α_r is a normalized attention weight.

Finally,f is obtained by performing a weighted summation of these regions, which f encompasses the contextual information of all regions. The context vector u_s is randomly initialized during training and jointly trained with the model.

$$f = \sum_{H \times W} \alpha_r V_r \tag{7}$$

Since the multi-head attention module has three parallel contextual attention modules, three feature vectors f_1, f_2, f_3 can be obtain.

A layer attention module is added after three parallel context-aware attention modules. The input to the layer attention module includes the feature map Z_s from the last convolutional layer of the style encoder and the feature vectors f_1, f_2, f_3 from the three parallel context-aware attention modules. Similarly, we feed the feature map Z_s into a single-layer MLP, followed by applying the softmax function to obtain three normalized scores $\varphi_1, \varphi_2, \varphi_3$. These normalized scores can be used to establish the relationship between regions and questions, indicating which region's features the model should focus on more.

$$\varphi_1, \varphi_2, \varphi_3 = softmax(\tanh(w_l Z_s + b_l)) \tag{8}$$

where w_l and b_l is the parameter of full connected layer.

$$Z = \sum_{i=1}^{3} \varphi_i f_i \tag{9}$$

where Z is the weighted sum of the three feature vectors. Since the style encoder takes k style images as input, the feature vector Z needs to be averaged to get Z_{ss}.

$$Z_{ss} = \frac{1}{k} \sum_{k} Z^k \tag{10}$$

2) CONTENT SELF-ADAPTATION MODULE

It is essential to preserve the semantic structure of the content image for the Chinese character generation task. To achieve this, this paper introduces a content self-adaptation module. The content self-adaptation module encodes the contextual information of the content image into local features and establishes corresponding contextual relationships on these local features to enhance the representational capacity of the semantic structure of the content image. In the following, we provide a detailed description of the structure of the content self-adaptation module.

As shown in Fig.3, the green parts represent the content self-adaptation module. Given a content feature map $Z_c \in \mathbb{R}^{C \times H \times W}$, whitening transformation [23] can remove irrelevant style and texture information from the content feature map, resulting in a new feature map $\tilde{Z}_c$. Subsequently, the whitened feature map $\tilde{Z}_c$ is passed through two convolutional layers to generate two new features $\tilde{Z}_{c1}$ and $\tilde{Z}_{c2}$, which are reshaped to $\mathbb{R}^{C \times N}$, where $N = H \times W$. Then, the transpose of $\tilde{Z}_{c1}$ is matrix multiplied with $\tilde{Z}_{c2}$.

Finally, the softmax function is used to compute the spatial attention map $S \in \mathbb{R}^{C \times N}$.

$$S_{ji} = softmax(\tilde{Z}_{ic1}^T \otimes \tilde{Z}_{jc2}) \tag{11}$$

where the symbol $\otimes$ presents matrix multiplication, and S_{ji} measures the mutual influence between the i-th position and the j-th position. The closer the features of two positions are, the stronger their correlation. This means that if two positions have similar features, it indicates a higher degree of correlation between them.

Meanwhile, the content feature map $Z_c \in \mathbb{R}^{C \times H \times W}$ is fed into another convolutional layer to obtain a new feature map $\tilde{Z}_{c3}$, which is reshaped to $R \in \mathbb{R}^{C \times N}$. The transpose of $\tilde{Z}_{c3}$ and S_{ji} performs matrix multiplication, and then Z_c is used for element-wise addition.

$$Z_{cc} = \tilde{Z}_{c3} \otimes S_{ji}^T + Z_c \tag{12}$$

3) CO-ADAPTATION MODULE

Through the multi-head attention module and the content self-adaptation module, we obtain the respective style and content features. Then, a co-adaptation module is used to calculate the correlation between the style and content features and recombine them into stylized new features.

As shown in Fig.3, the blue part represents the co-adaptation module, which has a similar structure to the content self-adaptation module. To fully utilize the long-range information captured by the multi-head attention module and the content self-adaptation module, the co-adaptation module fuse the features obtained from both modules. First, the features Z_{cc} and Z_{ss} are converted into $\tilde{Z}_{cc}$ and $\tilde{Z}_{ss}$ through the whitening operation. Then, $\tilde{Z}_{cc}$ and $\tilde{Z}_{ss}$ are input into two convolutional layers respectively to obtain two new features $\tilde{Z}_{cc1}$ and $\tilde{Z}_{ss2}$. Similarly, $\tilde{Z}_{cc1}$ is matrix-multiplied with the transpose of $\tilde{Z}_{ss2}$, and then the softmax function is applied to calculate the spatial mapping $A_{cs} \in \mathbb{R}^{N \times N}$ between them.

$$A_{cs} = softmax(\tilde{Z}_{cc1} \otimes \tilde{Z}_{ss2}^T) \tag{13}$$

At the same time, Z_{ss} is passed through an additional convolutional layer to obtain Z_{ss3}. Subsequently, a matrix multiplication is performed between Z_{ss3} and the transpose of A_{cs}. Finally, the result of the matrix operation is added element-wise to the content feature map Z_{cc} to obtain the final output Z_{cs}.

This process can be defined as:

$$Z_{cs} = A_{cs}^T \otimes Z_{ss3} + Z_{cc} \tag{14}$$

Then, Z_{cs} is concatenated with the style feature map Z_{ss} and fed into the decoder. The decoder generates a font image of size 64 × 64 from Z_{cs} and style feature map Z_{ss}.

C. DISCRIMINATOR

The discriminator distinguishes whether a font image is the ground truth or a fake font generated by the generator. To generate more realistic font images, a patch-level discriminator consisting of 3 convolutional layers is used in the discriminator. Simultaneously, the Adam optimizer [54] is used to update the parameters of the generator and discriminator. Our discriminator consists of a style discriminator and a content discriminator. The style discriminator assesses the stylistic similarity between the generated images and the real images, while the content discriminator determines whether the generated images from the generator have the same content as the real images. Similar to CycleGAN and DualGAN [55], our discriminator adopts the patchGAN [5] structure. The traditional GAN discriminator outputs a single True or False value, providing an overall evaluation of the generated images. In contrast, the patchGAN is inherently a fully convolutional network architecture that partitions the input image into multiple $N * N$ regions and makes discriminations for each region individually. The discriminator's output is the average of these patch-wise evaluations. By utilizing the patchGAN discriminator, we can reduce the size of the input image, decrease computational complexity, and better focus on local features.

D. LOSS FUNCTION

Up to this point, our model is essentially constructed. It is based on the Generative Adversarial Network (GAN) framework and is referred to as the generator G. Additionally, our model incorporates two types of loss functions during training: (1) Adversarial loss [56], which is used to train our model by solving a minimax problem, enabling Chinese font style transfer. (2) $\mathcal{L}_1$ loss, which is employed to stabilize the training of the model.

1) ADVERSARIAL LOSS

This work uses a standard adversarial game to train the generator G and the discriminator D of our Chinese character generation task. The generator G generates realistic but fake images in an attempt to deceive the discriminator. When both the generated fake images and real images are fed into the discriminator, the adversarial loss penalizes incorrect judgments, thereby enhancing the model's ability to generate convincing font images. Our adversarial loss consists of two components: one from the loss between the generator G and the style discriminator, and the other from the loss between the generator G and the content discriminator. Here, $E_{I_c \in P_c, I_s \in P_s}[\log(1 - D_s(\tilde{x}))]$ and $E_{I_c \in P_c, I_s \in P_s}[\log(1 - D_c(\tilde{x}))]$ are used to update the generator G, and $E_{I_c \in P_c, I_s \in P_s}[\log D_s(I_s)]$ is used to update the style discriminator. $E_{I_c \in P_c, I_s \in P_s}[\log D_c(I_c)]$ is used to update the content discriminator.

In summary, our model generates convincing font images through a minimax optimization process.

$$\mathcal{L}_{advs} = \underset{D_s}{\max}\underset{G}{\min} E_{I_c \in P_c, I_s \in P_s}\left[\log D_s(I_s) + \log(1 - D_s(\tilde{x}))\right] \tag{15}$$

$$\mathcal{L}_{advc} = \underset{D_c}{\max}\underset{G}{\min} E_{I_c \in P_c, I_s \in P_s}\left[\log D_c(I_c) + \log(1 - D_c(\tilde{x}))\right] \tag{16}$$

$$\mathcal{L}_{adv} = \mathcal{L}_{advc} + \mathcal{L}_{advs} \tag{17}$$

where $D_s(*)$ and $D_c(*)$ represent the outputs of the style discriminator and the content discriminator, respectively. $\tilde{x}$ is the real image, I_c is the content image, and I_s is the style image.

2) $\mathcal{L}_1$ LOSS

To ensure stable training of the model and encourage the generator G to generate output images that are similar to the real images, this article uses $\mathcal{L}_1$ loss to constrain the training of the model. The final generated images by the generator G should preserve both content C and style S.

$$\mathcal{L}_1 = E_{x,\tilde{x} \in P_{(x,\tilde{x})}} \|x - \tilde{x}\|_1 \tag{18}$$

3) FULL OBJECTIVE

Finally, we train the model using the following overall loss function:

$$\mathcal{L} = \lambda_{adv}\mathcal{L}_{adv} + \lambda_1\mathcal{L}_1 \tag{19}$$

where λ_{adv} and λ_1 are hyperparameters that can be adjusted during training to control the weights of the respective losses.

IV. EXPERIMENTAL RESULTS AND COMPARISON

This section introduces the dataset, experimental setup, and evaluation metrics. Then, the performance of the model is evaluated through a series of evaluations, and the experimental results are quantitatively and qualitatively analyzed. Ablation experiments were also performed simultaneously. Finally, the effectiveness of the proposed model in generating unknown language fonts is verified.

A. DATASETS AND EVALUATION METRICS

1) DATASETS

We choose Chinese and Latin alphabets with significant structural differences as language pairs to train our font generation model and validate our Chinese character generation method. Figure 5 shows a few examples of the dataset. The dataset consists of 847 grayscale fonts, each with about 1000 Chinese characters of the same style and 52 Latin letters of the same style. To maintain consistency, all glyph images are resized to 64 × 64 pixels and binarized. We used Microsoft Yahei as input for the content images and kept it fixed throughout the training and testing process. The content image is only used to index the class of the synthesized characters, so it can be replaced by other font styles. The style image, on the other hand, is used as input for 6 randomly selected letters from the 52 Latin letters. The training set consists of 818 fonts, which is expanded from FTransGAN [52], and is denoted as the Seen Fonts Seen Characters ($SFSC$) set. In order to validate the proposed method, we evaluated the generative power of the model on two test set: one with 29 unseen fonts and 1000 seen characters each, denoted as Unseen Fonts Seen Characters ($UFSC$), and the other with 818 seen fonts and 29 unseen characters each, denoted as Unseen Characters Seen Fonts ($UCSF$). Where unseen content and unseen fonts do not appear during training.

In order to verify the performance of the model in one-to-many languages, we also constructed a multi-language test set. The multilingual test set contains 10 fonts in 5 languages including Japanese, Korean, Greek, Chinese and Cyrillic, recorded as unseen fonts unseen characters ($UFUC$).

2) EVALUATION METRICS

Font styles are defined by local fine-grained shapes (e.g., strokes, sizes, etc.), leading to the possibility of multiple glyph variants similar to the target font in the generated fonts. Therefore it is challenging to use a unified metric to evaluate the performance of the Chinese character generation task. To address this problem, this paper uses a variety of pixel-level evaluation metrics (e.g., L1 loss, SSIM, MS-SSIM, etc.) to evaluate the similarity between generated fonts and ground truth fonts. In order to comprehensively evaluate the performance of the model, this paper uses the *Fréchet Inception Distance* (FID) and accuracy to evaluate the proposed method from the perspective of feature distance. Specifically, this paper trained two ResNet-50 networks [57] to assess the content and style of fonts, including content accuracy, style accuracy, the *Fréchet Inception Distance* (FID) scores for content, and the *Fréchet Inception Distance* (FID scores) for style.

B. EXPERIMENTAL DETAILS

Our model is implemented in PyTorch and is trained using an Nvidia RTX 3090 Ti GPU. The proposed model was built on pix2pix, thus some of the basic settings for the experiment follow those of pix2pix [5]. We set the values of λ_{adv} and λ_1 in the total loss function to 1 and 100, respectively. We use the Adam optimizer with a batch size of 256 to train the Chinese character generation model for 20 epochs. The learning rate is set to 0.0002 for the first 10 epochs and gradually decays to 0 for the remaining 10 epochs.

To mitigate the potential issue of overfitting, the model undergoes preprocessing during training, including operations such as rotation, scaling, and translation. These measures aim to enhance the model's generalization capability. Secondly, dropout is used in the generator to reduce the model's dependence on specific neurons, thereby preventing the model from overfitting. Finally we add some slight random noise in the style code Z_s.

C. EXPERIMENTAL RESULT AND COMPARISON METHODS

This section shows the experimental results of this work and analyzes them qualitatively and quantitatively with other models.

1) EXPERIMENTAL RESULT

This experiment uses the dataset mentioned in Section IV(A) and trains the model following the experimental details described in Section IV-B. Subsequently, we randomly select three different fonts from the generated results for demonstration. As shown in Fig.4, the first, fifth, and ninth rows represent the style reference images. Rows 3, 7 and 11 are our results. The fourth row, eighth row and twelfth row represent the ground truth.

From Fig.4, the generated images exhibit clear stroke structures and showcase detailed nuances at the ends of the strokes similar to the ground truth. Moreover, the generated results successfully retain the structural integrity of the characters while effectively incorporating stylistic features. The results show that the proposed model has a good learning ability.

The loss function curve of the proposed method are shown in Fig.7. It is not difficult to find from (b) that the loss value

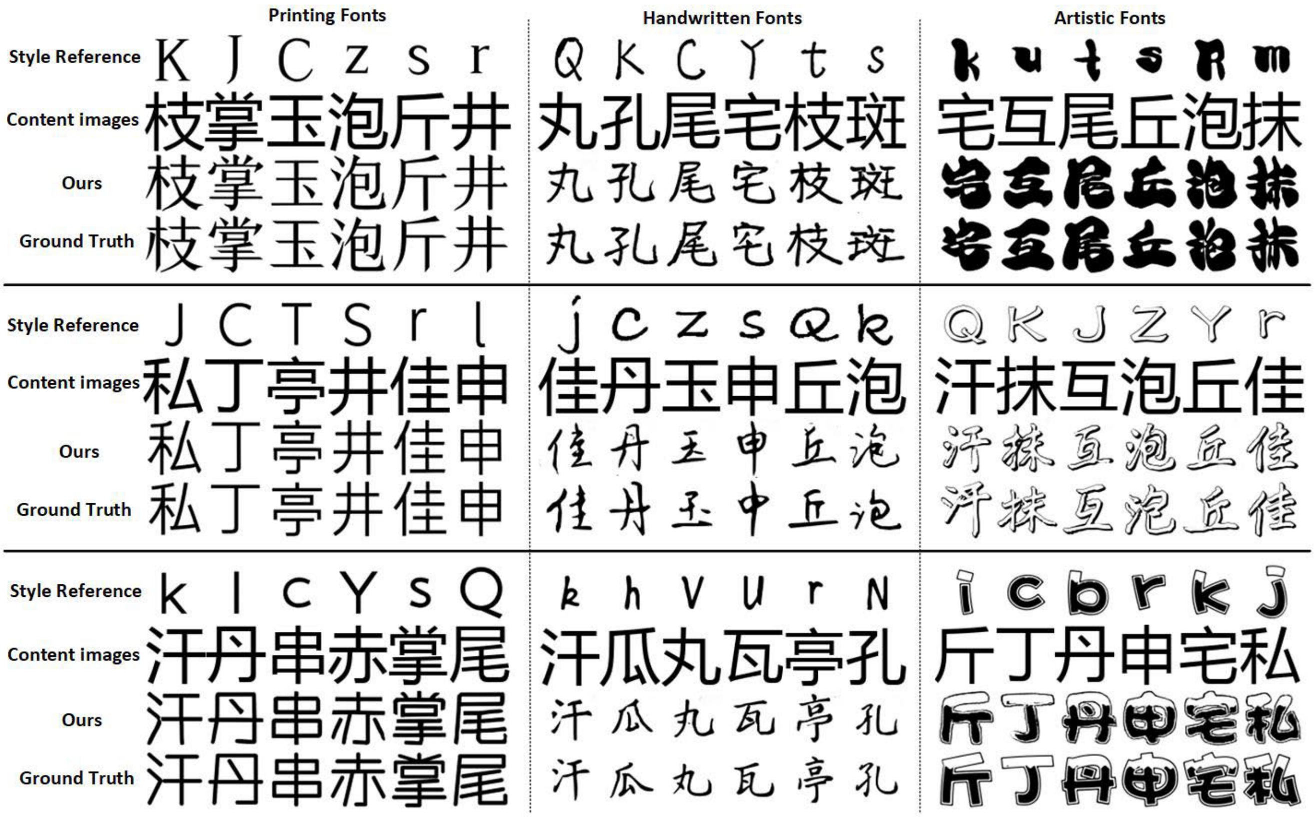

FIGURE 4. Partial presentation of experimental results. We present generated results with significant style variations, including printed fonts, handwritten fonts, and artistic fonts.

TABLE 2. Quantitative evaluation of the test set. Bold indicates the best, while underline indicates the second best.

Method	Unseen Characters							Unseen Fonts						
	L1 loss↓	SSIM↑	MS-SSIM↑	FID(c)↓	ACC(c)↑	FID(s)↓	ACC(s)↑	L1 loss↓	SSIM↑	MS-SSIM↑	FID(c)↓	ACC(c)↑	FID(s)↓	ACC(s)↑
DFS	0.174	0.328	0.257	269.347	0.7614	825.671	0.0191	0.202	0.258	0.249	327.420	0.7832	687.485	0.0064
MF-Net	0.140	0.468	0.296	161.666	0.8400	820.890	0.0046	0.210	0.272	0.229	278.012	0.7933	932.249	0.0004
FTransGAN	0.123	0.496	0.488	50.935	0.9723	318.729	0.5575	0.185	0.354	0.364	101.136	0.9979	452.298	0.0860
ours	**0.122**	**0.501**	**0.493**	**46.913**	**0.9726**	**306.343**	**0.5907**	**0.177**	**0.376**	**0.392**	**96.629**	**0.9987**	**429.847**	**0.1235**

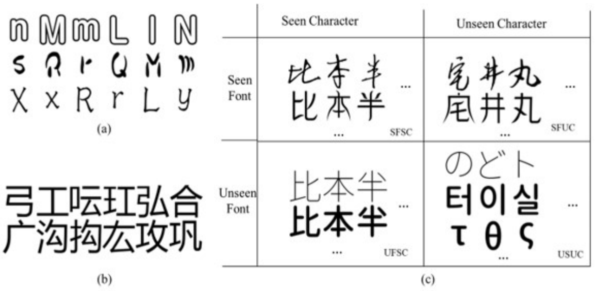

FIGURE 5. Sample example of data set. (*a*) are several style images from different styles, (*b*) are several content images from Microsoft Yahei, (*c*) are ground truth images.

of $\mathcal{L}_1$ first decreases and finally converges in the range of 26. In the adversarial loss function $\mathcal{L}_{adv}$, the generator fluctuates in a relatively large range and finally converges in the range of 1.5, while the style discriminator and content discriminator also gradually become stable.

2) COMPARISON METHODS

In this section, we compare our model with existing cross-lingual font generation methods: (1) FTransGAN [52] introduces contextaware modules and hierarchical attention modules to capture local and global features. (2) MF-Net is built upon the FTransGAN framework and incorporates a language complexity-aware skip connection to adjust character clarity. (3) DFS [58] stylizes the target font by decoding weighted deep features.All three models employ the idea of disentangling content and style and accomplish the task of cross-lingual font generation.

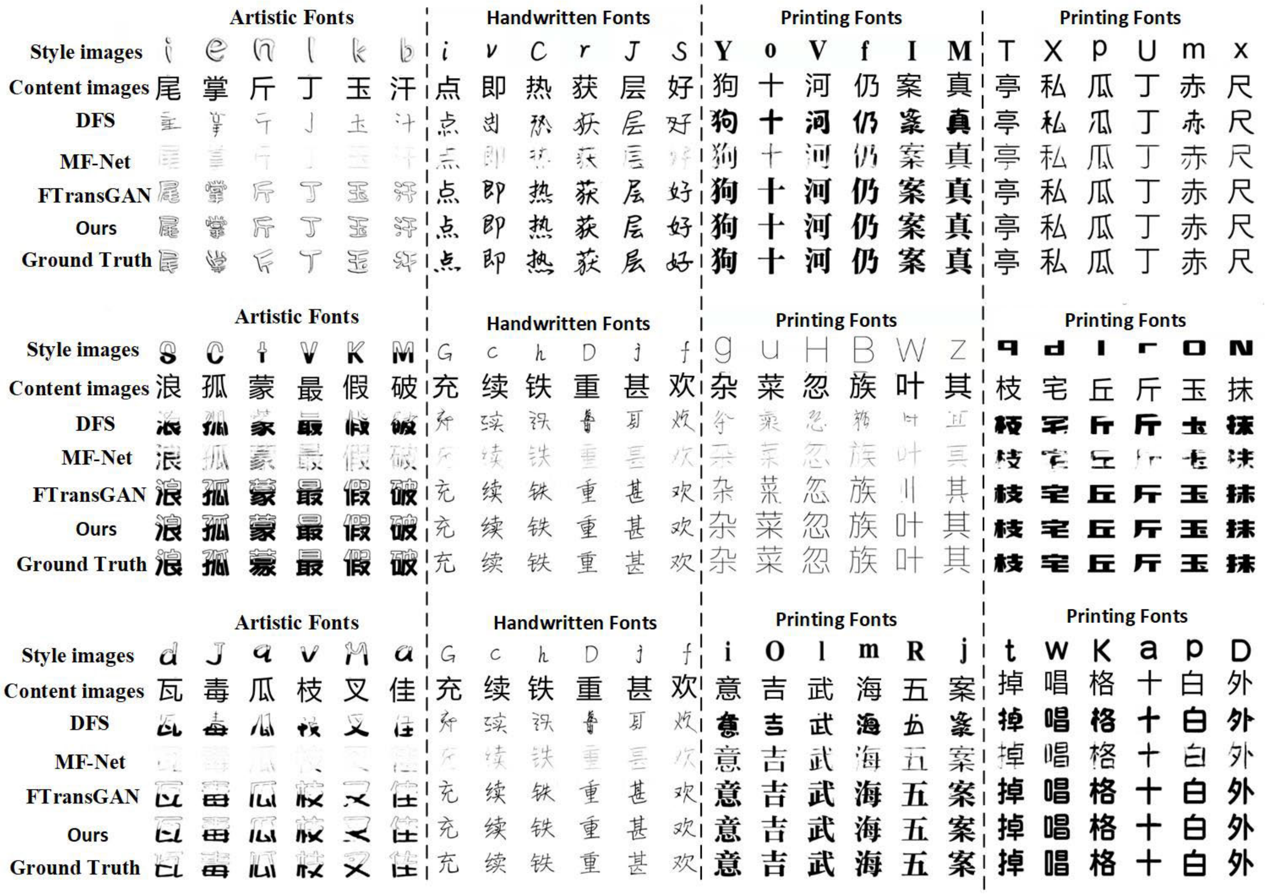
Artistic Fonts
Handwritten Fonts
Printing Fonts
Printing Fonts
Style images
Content images
DFS
MF-Net
FTransGAN
Ours
Ground Truth
尾 掌 斤 丁 玉 汗 点 即 热 获 层 好 狗 十 河 仍 案 真 亭 私 瓜 丁 赤 尺
浪 孤 蒙 最 假 破 充 续 铁 重 甚 欢 杂 菜 忽 族 叶 其 枝 宅 丘 斤 玉 抹
瓦 毒 瓜 枝 叉 佳 充 续 铁 重 甚 欢 意 吉 武 海 五 案 掉 唱 格 十 白 外

FIGURE 6. Qualitative comparison with other existing cross-language conversion models. In the experiments, We have selected several different fonts for display, including printing font, artistic font, and handwritten font.

a: QUANTITATIVE COMPARISON

Table 2 presents the performance comparison of our model with other few-shot font generation methods. To ensure fairness, we train all models using the dataset mentioned in Section IV(A) and evaluate them on both the Unseen Font Seen Characters and Unseen Characters Seen Font test sets. From Table 2, it can be observed that our model achieves the best overall performance, indicating its strong competitiveness compared to other models, particularly in predicting unknown styles. This implies that our model is capable of generating high-quality and diverse results for stylized font generation tasks on unseen fonts.

In addition, the accuracy of the classification results and the FID score are also visually displayed in the form of histograms. Two ResNet-50 networks are trained as classification models to evaluate different font generation methods by considering their ability to preserve content structure and style transfer. We use the various cross-language font generation methods mentioned earlier to generate stylized images. Subsequently, these stylized images generated by different methods are input into the pre-trained content and style classification network to obtain classification accuracy. The content classifier is used to distinguish which character the generated image belongs to, while the style classifier is used to distinguish which font style the generated image belongs to. A high accuracy in style classification indicates that the model can effectively capture meaningful style features from the stylized images. Similarly, a high accuracy in content classification suggests that the model can maintain the original character structure.

It can be concluded from Fig.9 that MF-Net and DFS have the worst accuracy in content and style, and cannot generate attractive fonts. Our method shows high accuracy in both content and style, which shows that our network establishes a balance between content and style. The content accuracy and style accuracy of the proposed method in unknown fonts are 99.87% and 12.35%, respectively. The content accuracy and style accuracy in unknown characters are 97.26% and 59.07%, respectively. This shows that our model has better overall performance in content and style classification.

The *Fréchet Inception Distance* (FID) is used to assess the distance between real images and generated images,

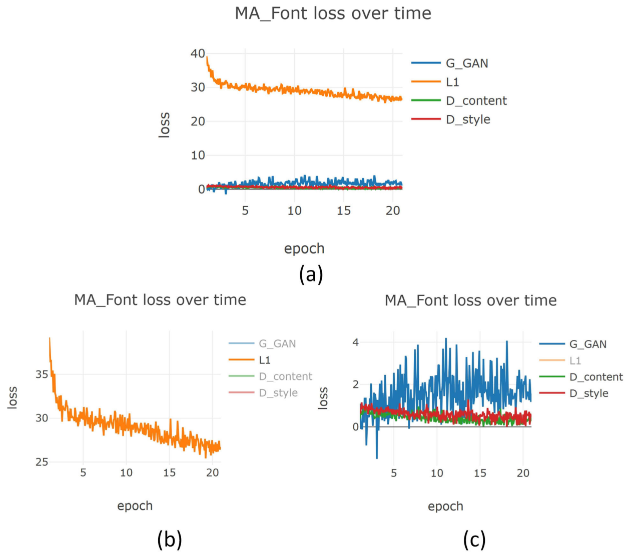

FIGURE 7. The loss function curve of the discriminator and generator during the training process. (a) shows all the loss function curves, (b) is the $\mathcal{L}_1$ loss function curve, (c) is the $\mathcal{L}_{adv}$ adversarial loss function curve.

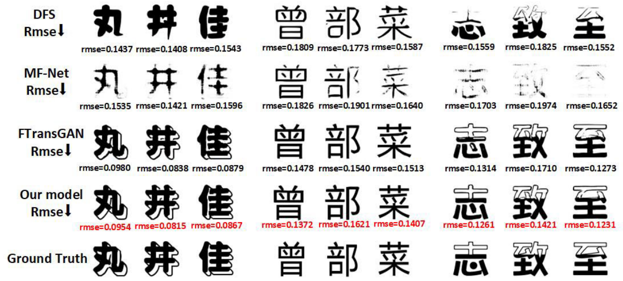
DFS Rmse⬇
rmse=0.1437 rmse=0.1408 rmse=0.1543
rmse=0.1809 rmse=0.1773 rmse=0.1587
rmse=0.1559 rmse=0.1825 rmse=0.1552
MF-Net Rmse⬇
rmse=0.1535 rmse=0.1421 rmse=0.1596
rmse=0.1826 rmse=0.1901 rmse=0.1640
rmse=0.1703 rmse=0.1974 rmse=0.1652
FTransGAN Rmse⬇
rmse=0.0980 rmse=0.0838 rmse=0.0879
rmse=0.1478 rmse=0.1540 rmse=0.1513
rmse=0.1314 rmse=0.1710 rmse=0.1273
Our model Rmse⬇
rmse=0.0954 rmse=0.0815 rmse=0.0867
rmse=0.1372 rmse=0.1621 rmse=0.1407
rmse=0.1261 rmse=0.1421 rmse=0.1231
Ground Truth
丸 井 佳 曾 部 菜 志 致 至

FIGURE 8. Visualization results of different methods. RMSE is a quantitative evaluation of each example in terms of content. Red color indicates the best RMSE.

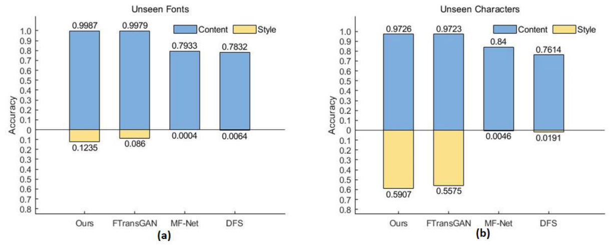

FIGURE 9. Comparison of classification accuracy of different font generation methods.

with smaller values indicating better model performance. We also use two classifiers as feature extractors and calculate *Fréchet Inception Distance*. As can be seen from Fig.10, the performance of our model is the best among several cross-language font generation methods.

b: QUALITATIVE COMPARISON

Fig.6 illustrates the results of qualitative comparison with the baseline methods. To evaluate the performance of our model in Chinese character generation tasks, Both the proposed method and the compared methods generate different samples on the datasets of unknown fonts and unknown characters. Twelve fonts with significant stylistic variations, including printed, handwritten, and artistic fonts, are selected for comparison with results generated by other methods. From the examples shown in Fig.6, it can be observed that our model produces realistic images that largely maintain consistency with the content images in terms of glyph structures while resembling the style reference fonts in font styles. However, for some fonts with shallow strokes, the DFS model can result in blurry effects. Similarly, when dealing with challenging artistic fonts and handwritten fonts, the results produced by the DFS model are relatively poorer. MF-Net designs a language complexity-aware skip connection to adjust the clarity of characters, but it often leads to blurry effects in the generated results. FTransGAN appears to capture detailed style nuances and generate complete

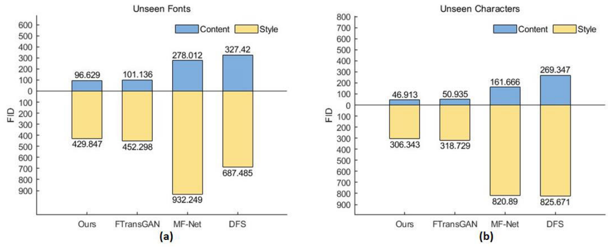

FIGURE 10. Comparison of FID Among different methods.

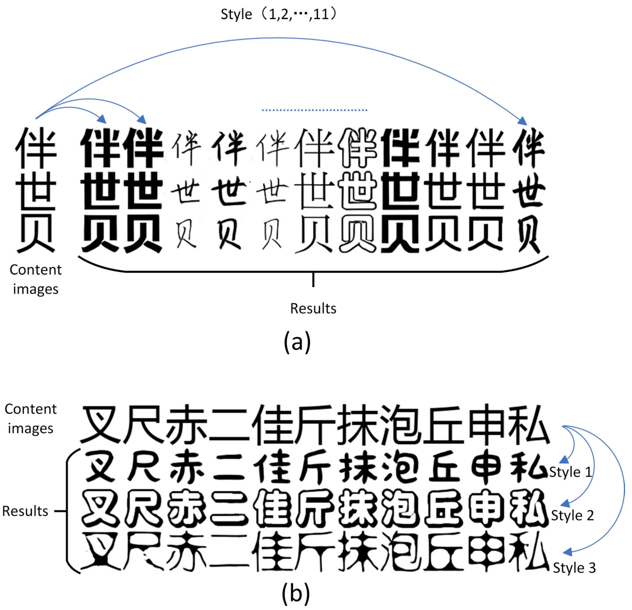

FIGURE 11. Validation experiments. (a) Effectiveness of Style Variables. Three different Chinese characters are used as reference content to generate fonts with unknown styles. Unknown style means that the image content is known during training, while unknown style is used for testing. (b) Validity of character variables. Generate fonts for unknown characters using three known style fonts as reference style. Unknown character refers to fonts whose style is known during training but unknown characters are used during testing.

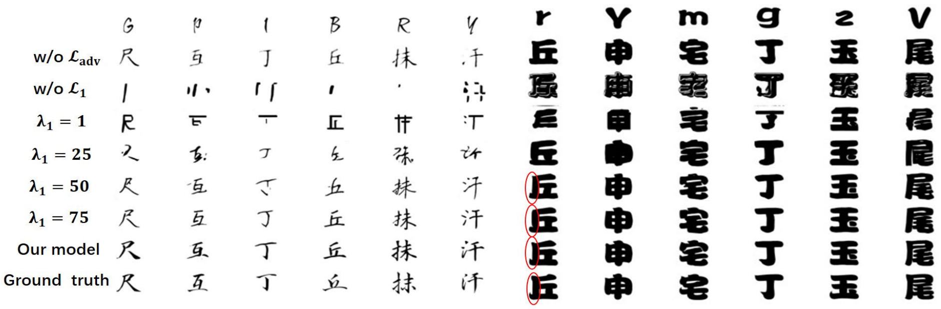

FIGURE 12. Visual samples of objective functions analysis. Our model shows the best overall performance.

TABLE 3. Ablation results for different modules. Bold indicates the best, while underline indicates the second best.We use C, N and S to denote the content self-adaptation module, the co-adaptation module, and the multi-head attention module, respectively. The model containing the three modules has the best combined performance. w/o means without.

Method	Unseen Characters							Unseen Fonts						
	L1 loss ↓	SSIM ↑	MS-SSIM ↑	FID(c) ↓	ACC(c) ↑	FID(s) ↓	ACC(s) ↑	L1 loss ↓	SSIM ↑	MS-SSIM ↑	FID(c) ↓	ACC(c) ↑	FID(s) ↓	ACC(s) ↑
w/o C, N and S	0.131	0.467	0.454	58.163	0.9689	408.669	0.3554	0.182	0.354	0.374	114.734	0.9953	450.676	0.0733
w/o N and S	0.126	0.486	0.475	<u>54.047</u>	0.9690	360.880	0.4553	<u>0.178</u>	<u>0.366</u>	<u>0.376</u>	104.938	<u>0.9976</u>	**404.874**	**0.1305**
w/o S	<u>0.125</u>	<u>0.488</u>	<u>0.477</u>	54.412	<u>0.9697</u>	<u>344.304</u>	<u>0.4996</u>	0.183	0.357	0.367	<u>105.720</u>	0.9974	<u>427.670</u>	0.1099
Full module	**0.122**	**0.501**	**0.493**	**46.913**	**0.9726**	**306.343**	**0.5907**	**0.177**	**0.376**	**0.392**	**96.629**	**0.9987**	429.847	<u>0.1235</u>

TABLE 4. Impact of objective functions. We report the effects of the adversarial loss and the L1 loss on the model, as well as analyzing the effects of different hyper-parameters λ. Our model is in the bottom row and it shows the best overall performance.

Method	Unseen Characters							Unseen Fonts						
	L1 loss ↓	SSIM ↑	MS-SSIM ↑	FID(c) ↓	ACC(c) ↑	FID(s) ↓	ACC(s) ↑	L1 loss ↓	SSIM ↑	MS-SSIM ↑	FID(c) ↓	ACC(c) ↑	FID(s) ↓	ACC(s) ↑
w/o $\mathcal{L}_{adv}$	**0.118**	<u>0.494</u>	<u>0.468</u>	92.784	0.8518	571.520	0.2950	**0.172**	<u>0.363</u>	<u>0.367</u>	166.824	0.8929	664.919	0.0327
w/o $\mathcal{L}_1$	0.282	0.148	0.020	430.521	0.2842	850.852	0.0166	0.300	0.097	0.004	521.329	0.4414	747.444	0.0243
$\lambda_1 = 1$	0.229	0.234	0.133	179.192	0.7517	865.733	0.0180	0.258	0.165	0.102	273.126	0.8150	764.832	0.0183
$\lambda_1 = 25$	0.173	0.344	0.292	96.348	0.9117	657.226	0.0836	0.208	0.264	0.246	165.262	0.9618	577.207	0.0169
$\lambda_1 = 50$	0.144	0.425	0.392	70.252	0.9474	530.019	0.2004	0.196	0.306	0.302	133.544	0.9886	556.882	0.0210
$\lambda_1 = 75$	0.135	0.457	0.436	<u>59.117</u>	**0.9727**	<u>416.580</u>	<u>0.3374</u>	0.189	0.335	0.340	<u>111.732</u>	<u>0.9914</u>	<u>495.834</u>	<u>0.0507</u>
Full module	<u>0.122</u>	**0.501**	**0.493**	**46.913**	<u>0.9726</u>	**306.343**	**0.5907**	<u>0.177</u>	**0.376**	**0.392**	**96.629**	**0.9987**	**429.847**	**0.1235**

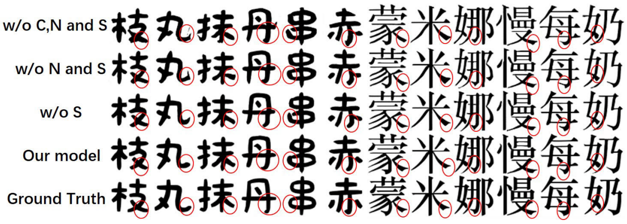

FIGURE 13. The qualitative results of the ablation experiments. C, N, and S represent the Content Self-adaptation Module, Co-adaptation Module, and Multi-Head Attention Module, respectively. The local details of the font are highlighted with red circles. As we sequentially added different modules, the generated font images became closer to the target font.

characters but overlooks fine-grained local styles such as handwritten fonts.

Fig.8 shows some of the visualization results of the different approaches, as well as the results of the RMSE evaluation for each example. Red color indicates the best RMSE.Our model shows the best RMSE, which indicates that the proposed method is closer to the ground truth images in terms of stroke structure.

D. VALIDATION EXPERIMENTS

In experiments, fonts with known characters or known styles are used to generate fonts with unknown styles or unknown characters (e.g., the first column in Figure 11 (a) or the second and third rows in Figure 11 (b)). The validation of these characters can strongly demonstrate the effectiveness of style variables and content variables.

1) EFFECTIVENESS OF STYLE VARIABLES

In the experiment, three known characters were randomly selected as content references, and then these three characters were used to generate fonts of 11 unknown styles. The final generated results are shown in Figure 11 (a), where the first column represents known characters, and the rest are fonts generated with unknown styles. From the generated results, it can be observed that fonts of unknown style can be generated convincingly with three different known characters. This indicates that the model possesses a strong capability to generate fonts with unknown styles.

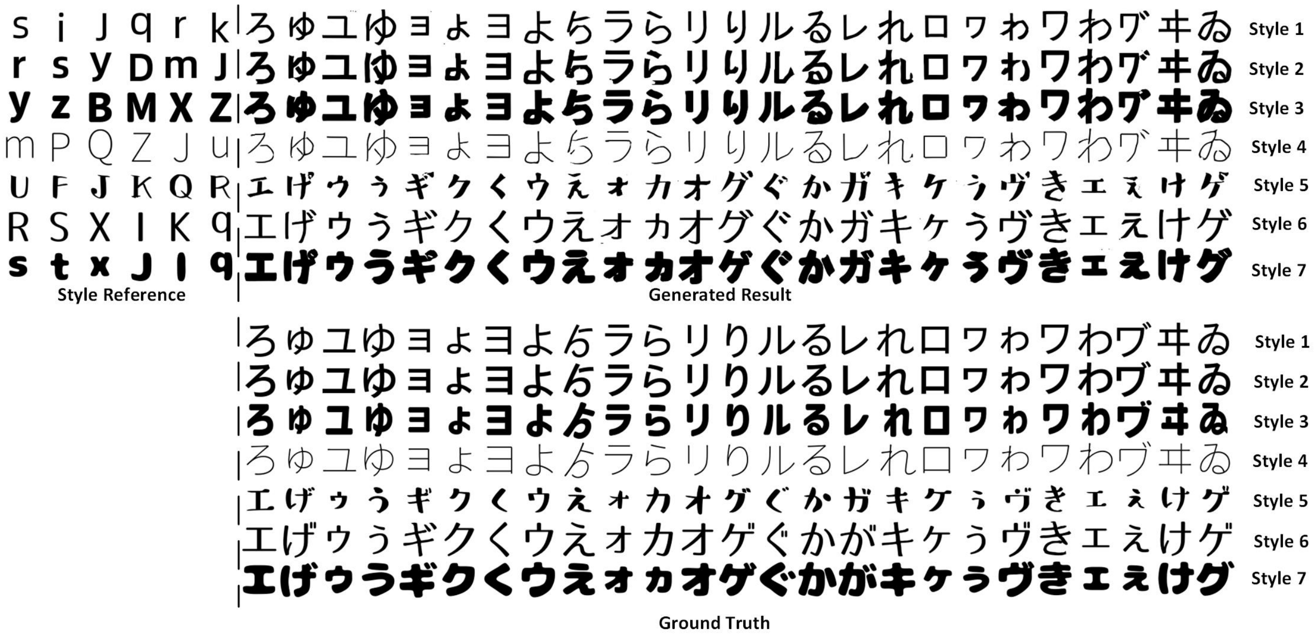

FIGURE 14. Generated results from the English-to-Japanese experiment. The picture shows 7 different styles, each with 25 characters.

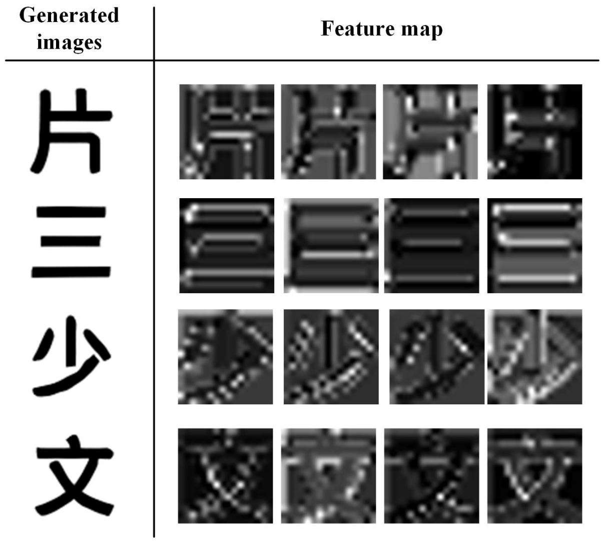

FIGURE 15. Visualization of multi-adaptation modules. The brighter areas are the style features that contribute more.

2) EFFECTIVENESS OF CONTENT VARIABLES

In the experiments, three different known styles of fonts are selected as style references. Initially, 11 unknown characters are input into the content encoder to obtain content features, which are then combined with the style references to generate the final results. From Fig.11 (b), it can be observed that the generated characters closely match the input characters. The content encoder is capable of extracting content image features comprehensively, unaffected by the stroke layout of characters. This indicates that content images are merely used to index the category of the characters one wishes to synthesize.

E. ABLATION STUDY

In order to verify the effectiveness of each module, this paper conducts ablation experiments on the proposed method. Specifically, we name the three modules C, S, and N, respectively. We first separate these modules and then add these modules to the model sequentially while keeping other settings unchanged. Table 3 reports the results of ablation experiments with multiple adaptation modules. The first row is the quantified result of the feature embedding that directly connects the two encoders, the second row is the quantified result of adding the content self-adaptation module, the third row is the quantified result of adding the content self-adaptation module and the co-adaptation module, and the The fourth rows are quantitative results for the full model. After adding modules in sequence, the values of SSIM, MS-SSIM and accuracy gradually increase, and the values of L1 loss and FID gradually decrease. Among them, the model containing three modules performs best. This shows that these modules can further improve the performance of the model.

Fig.13 shows the results of ablation study for these three modules. The local detail changes of each character are marked with red circles. As we add different modules sequentially, the generated font images become closer to the target font. The results show that the method is able to capture the local details of fonts and generate high-quality images similar to the target font.

In addition, the loss function is analyzed in this paper. Table 4 reports the relevant evaluation metrics for L_1 loss and adversarial loss, and a partial visualization of the results is shown in Fig.12. From Table 4, it can be seen that when there is no adversarial loss, the model does not show the best performance. When there is no L_1 loss, the worse the

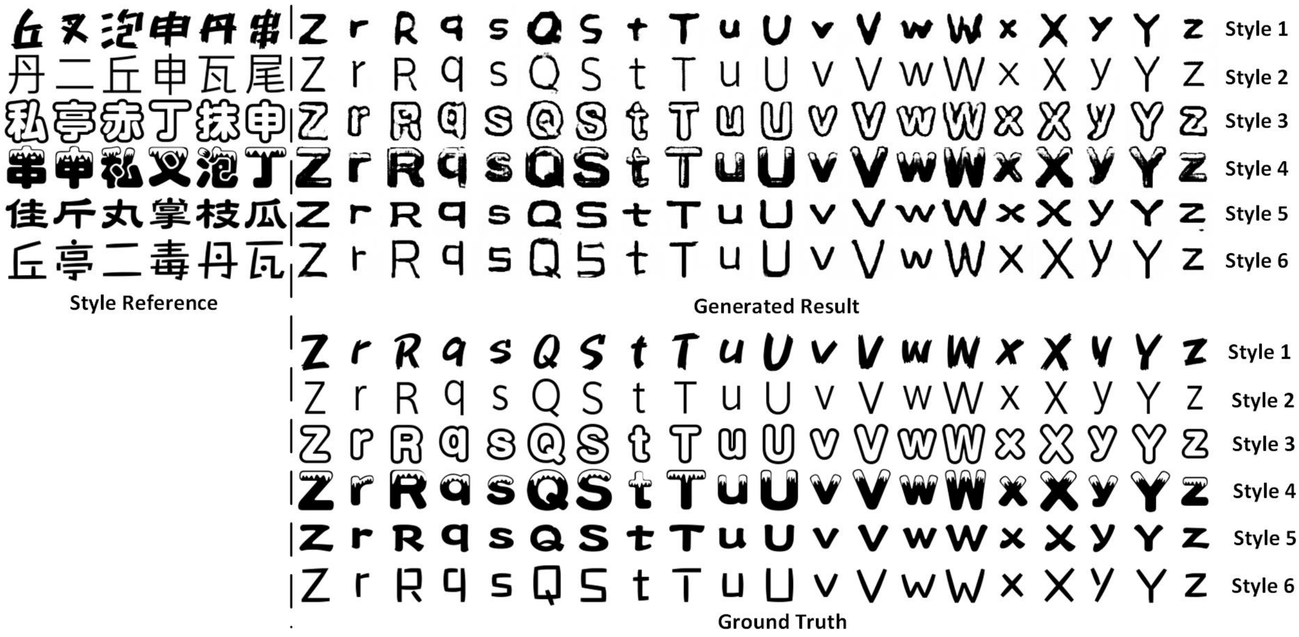

FIGURE 16. Generated results from the Chinese-to-English experiment. The picture shows 6 different styles, each with 20 characters.

evaluation value of the model, the very low quality of the generated images. The performance of the model gradually improves when increasing the value of the λ_1. The model which contains both adversarial loss and L_1 loss shows the best performance.

F. ANALYSIS FOR MULTI-ADAPTATION MODULE

To demonstrate the effectiveness of the multi-adaptation module, we visualize the feature mappings generated by the multi-adaptation module. Fig.15 shows the visualized image of the multi-adaptation module. From the figure, we can observe that the feature mappings Z_{cs} retain the semantic information of the content images well, which helps to generate well-structured characters. On the other hand, the style features can be reorganized based on the semantic information of the content features, and the brighter points in the figure are the reorganized style features.

The brighter regions in the figure indicate the style features that contribute more, and the co-adaptation module redistributes the style features to some localized regions that are easy to be ignored according to the distribution of the content features.

G. EXTENSION TO OTHER LANGUAGES

To further verify the validity and generalization ability of the model in other languages, we conducted one-to-one language and one-to-many language experiments. The unknown language test set is collected from free websites.

1) ONE-TO-ONE LANGUAGE

To verify the effectiveness of our proposed method on unknown languages, we use a test set consisting of 30 Japanese fonts. The proposed model requires random input of a fixed number of style images during the training phase. However, during the testing phase, the input for style images is unrestricted. Therefore, when synthesizing fonts for an unknown language during the testing phase, there is no need to retrain the model. This experiment evaluates the robustness of the model by introducing an unknown language as a distractor. Initially, the model is trained for style transfer from English to Chinese. Subsequently, Japanese font is employed as the unknown language test set. As can be seen from Fig.14, even if the model cannot see all fonts, it can learn the structural information of characters and the style information of style images very well. This indicates that the model possesses strong robustness.

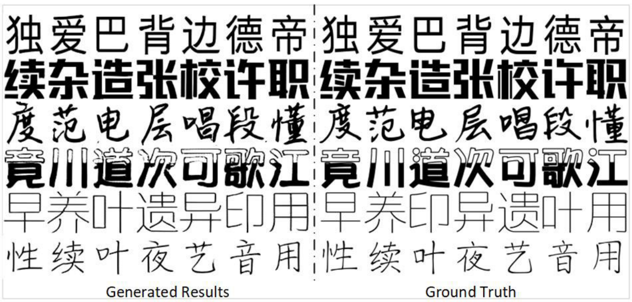

FIGURE 17. The generated fonts are the outcomes of Chinese-to-Chinese font generation. In Chinese-to-Chinese font generation experiments, using Chinese as input for content images and style images.

In addition, to demonstrate the model's capability for style transfer across languages, we also implement two experiments: "chinese2english" and "chinese2chinese". In the

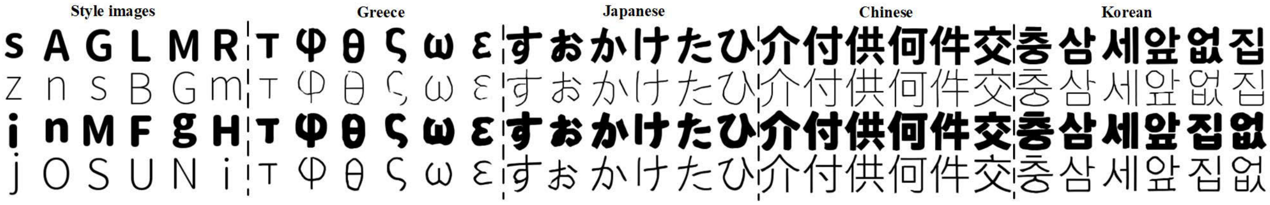

FIGURE 18. Experimental results generated by one-to-many languages. Generating four languages with the same style using the Latin alphabet as a style reference.

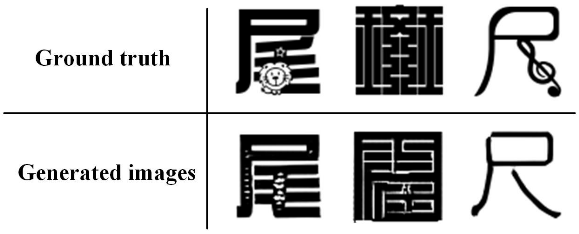

FIGURE 19. Failure case. The model does not perform well for complex characters with highly artistic and compact layouts.

"chinese2chinese" experiment, Chinese is used as input for content and style images. As seen in Fig.17, the synthesized font maintains the same style as Ground Truth, and the structure of the characters is clearly visible.

In the "chinese2english" experiment, we re-split the dataset to facilitate training and testing for "chinese2english". In this case, 818 fonts are utilized for training, 29 fonts are designated as unknown styles, and 6 letters are employed as unknown content. The experimental results for "chinese2english" are visible in Fig.16.

Our model can achieve style transfer from Chinese to English and from Chinese to Chinese. This shows that the model can be applied to font style migration in different situations.

2) ONE-TO-MANY LANGUAGES

It is challenging to only learn the style information of one language to generate fonts for two or more languages. Previous work mainly implemented one-to-many font style migration in the same language, so this paper collects a multi-language test set (UFUC) for one-to-many language style migration. Among them, Korean, Japanese and Cyrillic are unknown languages and they did not appear during the training process. Figure.18 reports the experimental results of one-to-many language conversion, the first column is the style image, and the remaining four columns are the generation results. It can be observed from the figure that the generated character structure is complete and conforms to human visual perception. However, there are some shortcomings in one-to-multi-language font style transfer. For example, the generated Japanese is unnatural.

H. FAILURE CASES AND LIMITATION

Figure 19 shows complex characters with highly artistic and compact layouts. The first and third characters have complete structures, but they ignore some highly artistic subtle patterns. The second image has very compact strokes that result in poor model performance.

V. CONCLUSION

This paper proposes an effective Chinese character generation model, and a large number of font generation experiments verify the effectiveness of the model. The 7 indicators measured on the unknown content known font test set and the unknown font known content test set are sufficient to illustrate the superiority of the model. Compared with existing cross-language fonts, our model achieves better results both qualitatively and quantitatively. An important role is played by the multi-adaptation module, which readjusts the distribution of style features. This shows that our model considers global content structure and local style features to generate high quality images.

In addition, we conducted visualization and ablation experiments on the multi-adaptation module to analyze its role in depth. The visualization results illustrate that the multi-adaptation module can distribute the style features in some local regions that are easily neglected, enhance the style migration effect of the model, and improve the model generation capability. The data from the ablation experiments illustrate that the multi-adaptation module is effective and it improves the model generation performance. We also analyzed the effect of the objective function on the model. The results show that the adversarial loss function and the L1 loss function are beneficial to stabilize the training of the model and improve the image quality. Then we analyze the effect of content variables and style variables on image quality. Finally we try to extend to style migration for one-to-many languages.

However, our model still has some shortcomings. We show several failure cases in Figure 19. The model cannot achieve style transfer for some highly artistic patterns and fonts with compact stroke distribution. Second, the model can only output fixed-size images. In the future, we should continue to improve the model's generation capabilities so that the model can adapt to highly artistic patterns and compact font styles. Secondly, the generated results are converted into vector font files, which is more convenient for practical applications.

REFERENCES

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ''Generative adversarial networks,'' *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[3] Y. Tian, ''Rewrite: Neural style transfer for Chinese fonts, 2016,'' *Retrieved*, vol. 23, pp. 1–14, Jul. 2016.

[4] Y. Tian. (2017). *Zi2ZI: Master Chinese Calligraphy With Conditional Adversarial Networks*. [Online]. Available: https://github.com/kaonashi-tyc/zi2zi

[5] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, ''Image-to-image translation with conditional adversarial networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[6] S.-J. Wu, C.-Y. Yang, and J. Yung-jen Hsu, ''CalliGAN: Style and structure-aware Chinese calligraphy character generator,'' 2020, *arXiv:2005.12500*.

[7] S. Yuan, R. Liu, M. Chen, B. Chen, Z. Qiu, and X. He, ''SE-GAN: Skeleton enhanced gan-based model for brush handwriting font generation,'' in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2022, pp. 1–6.

[8] J. Chang and Y. Gu, ''Chinese typography transfer,'' 2017, *arXiv:1707.04904*.

[9] P. Lyu, X. Bai, C. Yao, Z. Zhu, T. Huang, and W. Liu, ''Auto-encoder guided GAN for Chinese calligraphy synthesis,'' in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 1, Nov. 2017, pp. 1095–1100.

[10] S. Zeng and Z. Pan, ''An unsupervised font style transfer model based on generative adversarial networks,'' *Multimedia Tools Appl.*, vol. 81, no. 4, pp. 5305–5324, Feb. 2022.

[11] J. Chen, Y. Ji, H. Chen, and X. Xu, ''Learning one-to-many stylised Chinese character transformation and generation by generative adversarial networks,'' *IET Image Process.*, vol. 13, no. 14, pp. 2680–2686, Dec. 2019.

[12] B. Chang, Q. Zhang, S. Pan, and L. Meng, ''Generating handwritten Chinese characters using CycleGAN,'' in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 199–207.

[13] J. Zeng, Q. Chen, Y. Liu, M. Wang, and Y. Yao, ''StrokeGAN: Reducing mode collapse in Chinese font generation via stroke encoding,'' in *Proc. AAAI Conf. Artif. Intell.*, May 2021, pp. 3270–3277.

[14] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, ''Unpaired image-to-image translation using cycle-consistent adversarial networks,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, ''Densely connected convolutional networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[16] J. Zhou, Y. Wang, Y. Yuan, Q. Huang, and J. Zeng, ''SGCE-font: Skeleton guided channel expansion for Chinese font generation,'' 2022, *arXiv:2211.14475*.

[17] J. Zeng, Y. Wang, Q. Chen, Y. Liu, M. Wang, and Y. Yao, ''StrokeGAN+: Few-shot semi-supervised Chinese font generation with stroke encoding,'' 2022, *arXiv:2211.06198*.

[18] J. Cha, S. Chun, G. Lee, B. Lee, S. Kim, and H. Lee, ''Few-shot compositional font generation with dual memory,'' in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 735–751.

[19] Y. Gao, Y. Guo, Z. Lian, Y. Tang, and J. Xiao, ''Artistic glyph image synthesis via one-stage few-shot learning,'' *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–12, Dec. 2019.

[20] Y. Gao and J. Wu, ''GAN-based unpaired Chinese character image translation via skeleton transformation and stroke rendering,'' in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, pp. 646–653.

[21] D. Sun, T. Ren, C. Li, H. Su, and J. Zhu, ''Learning to write stylized Chinese characters by reading a handful of examples,'' 2017, *arXiv:1712.06424*.

[22] C. Li, Y. Taniguchi, M. Lu, S. Konomi, and H. Nagahara, ''Cross-language font style transfer,'' *Int. J. Speech Technol.*, vol. 53, no. 15, pp. 18666–18680, Aug. 2023.

[23] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, ''Universal style transfer via feature transforms,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–13.

[24] L. A. Gatys, A. S. Ecker, and M. Bethge, ''Image style transfer using convolutional neural networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.

[25] H. Kazemi, S. M. Iranmanesh, and N. Nasrabadi, ''Style and content disentanglement in generative adversarial networks,'' in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Iran, Jan. 2019, pp. 848–856.

[26] D. Kotovenko, A. Sanakoyeu, S. Lang, and B. Ommer, ''Content and style disentanglement for artistic style transfer,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4421–4430.

[27] X. Yu, Y. Chen, S. Liu, T. Li, and G. Li, ''Multi-mapping image-to-image translation via learning disentanglement,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–6.

[28] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, ''Few-shot unsupervised image-to-image translation,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10550–10559.

[29] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, ''Dual attention network for scene segmentation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3141–3149.

[30] Y. Deng, F. Tang, W. Dong, W. Sun, F. Huang, and C. Xu, ''Arbitrary style transfer via multi-adaptation network,'' in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2719–2727.

[31] X. Huang and S. Belongie, ''Arbitrary style transfer in real-time with adaptive instance normalization,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1510–1519.

[32] V. Kitov, ''Real-time style transfer with strength control,'' in *Proc. 18th Interface Conf.*, 2019, pp. 206–218.

[33] H. Liu, T. Liu, Z. Zhang, A. K. Sangaiah, B. Yang, and Y. Li, ''ARHPE: Asymmetric relation-aware representation learning for head pose estimation in industrial human–computer interaction,'' *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 7107–7117, Oct. 2022.

[34] D. Bahdanau, K. Cho, and Y. Bengio, ''Neural machine translation by jointly learning to align and translate,'' 2014, *arXiv:1409.0473*.

[35] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, ''Hierarchical attention networks for document classification,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2016, pp. 1480–1489.

[36] D. Yu, J. Fu, T. Mei, and Y. Rui, ''Multi-level attention networks for visual question answering,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4187–4195.

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–12.

[38] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, ''Self-attention generative adversarial networks,'' in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7354–7363.

[39] D. Y. Park and K. H. Lee, ''Arbitrary style transfer with style-attentional networks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5873–5881.

[40] Y. Deng, F. Tang, W. Dong, H. Huang, C. Ma, and C. Xu, ''Arbitrary video style transfer via multi-channel correlation,'' in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2021, pp. 1210–1217.

[41] Y. Jiang, Z. Lian, Y. Tang, and J. Xiao, ''DCFont: An end-to-end deep Chinese font generation system,'' in *Proc. SIGGRAPH Asia Tech. Briefs*, no. 4, Bangkok, Thailand. New York, NY, USA: Association for Computing Machinery, 2017, Art. no. 22, doi: 10.1145/3145749.3149440.

[42] Y. Jiang, Z. Lian, Y. Tang, and J. Xiao, ''SCFont: Structure-guided Chinese font generation via deep stacked networks,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 4015–4022.

[43] W. Liu, F. Liu, F. Ding, Q. He, and Z. Yi, ''XMP-Font: Self-supervised cross-modality pre-training for few-shot font generation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7895–7904.

[44] C. Wang, M. Zhou, T. Ge, Y. Jiang, H. Bao, and W. Xu, ''CF-Font: Content fusion for few-shot font generation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 1858–1867.

[45] M. Yao, Y. Zhang, X. Lin, X. Li, and W. Zuo, ''VQ-Font: Few-shot font generation with structure-aware enhancement and quantization,'' 2023, *arXiv:2308.14018*.

[46] X. He, M. Zhu, N. Wang, X. Gao, and H. Yang, ''Few-shot font generation by learning style difference and similarity,'' 2023, *arXiv:2301.10008*.

[47] M. Qin, Z. Zhang, and X. Zhou, ''Disentangled representation learning GANs for generalized and stable font fusion network,'' *IET Image Process.*, vol. 16, no. 2, pp. 393–406, Feb. 2022.

[48] Y. Zhang, Y. Zhang, and W. Cai, ''A unified framework for generalizable style transfer: Style and content separation,'' *IEEE Trans. Image Process.*, vol. 29, pp. 4085–4098, 2020.

[49] S. Park, S. Chun, J. Cha, B. Lee, and H. Shim, ''Multiple heads are better than one: Few-shot font generation with multiple localized experts,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13880–13889.

[50] Y. Xie, X. Chen, L. Sun, and Y. Lu, "DG-Font: Deformable generative networks for unsupervised font generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5126–5136.

[51] S. Park, S. Chun, J. Cha, B. Lee, and H. Shim, "Few-shot font generation with localized style representations and factorization," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 3, pp. 2393–2402.

[52] C. Li, Y. Taniguchi, M. Lu, and S. Konomi, "Few-shot font style transfer between different languages," in *Proc. IEEE/CVF winter Conf. Appl. Comput. Vis.*, 2021, pp. 433–442.

[53] Y. Zhang, J. Man, and P. Sun, "MF-Net: A novel few-shot stylized multilingual font generation method," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 2088–2096.

[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[55] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2868–2876.

[56] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[58] A. Zhu, X. Lu, X. Bai, S. Uchida, B. K. Iwana, and S. Xiong, "Few-shot text style transfer via deep feature similarity," *IEEE Trans. Image Process.*, vol. 29, pp. 6932–6946, 2020.

YANBO QIU is currently pursuing the degree with Changzhou University, Jiangsu, China. His major is circuits and systems. His current research interests include image processing and computer vision.

KAIBIN CHU received the M.E. degree from Changzhou University. He is currently a Professor with Changzhou University. He is also mainly engaged in the research of applied electronics technology. At present, the projects, he has completed or is in the process of completing are the research of automatic battery testing technology, the research of withstand voltage testing equipment, and the research of impedance testing projects.

JI ZHANG received the M.E. degree from Nanjing University of Science and Technology. He is currently an Associate Professor with Changzhou University. His main research interests include image processing and machine vision.

CHENGTAO FENG received the Ph.D. degree from Nanjing University of Aeronautics and Astronautics. His main research interests include indoor positioning and inertial vision odometry.