Diversity Regularized StarGAN for Multi-style Fonts Generation of Chinese Characters

Jinshan Zeng\textsuperscript{1,a}, Qi Chen\textsuperscript{1,b}, Mingwen Wang\textsuperscript{1,*}

\textsuperscript{1} School of Computer and Information Engineering and Institute of Artificial Intelligence, Jiangxi Normal University, Nanchang, China

E-mail: \textsuperscript{a}jinshanzeng@jxnu.edu.cn, \textsuperscript{b}chenqi970226@gmail.com

Corresponding author: \textsuperscript{*}mwang@jxnu.edu.cn

Abstract. The generation of stylish Chinese fonts plays a central role in many applications such as the design of art fonts and Chinese calligraphy generation. Most of existing methods focus on the generation of a single-style Chinese font, while few works focus on the multi-style font generation. In this paper, we exploit the star generative adversarial networks (StarGAN), a very popular generative adversarial networks (GAN) model recently developed in the literature, to realize the generation of multi-style Chinese fonts via a single model. Furthermore, in order to tackle the generation issue of Chinese characters having similar strokes for StarGAN, i.e., generating the same mode for these different but similar Chinese characters, we introduce a diversity regularizer such that the generator can generate high-quality characters with better diversity. A series of experiments are conducted on a handwritten Chinese character dataset called \textit{CASIA-HWDB1.1} and three standard printing font datasets to show the effectiveness of the proposed method. The experiment results show that the proposed method can effectively tackle the generation issue of Chinese characters having similar strokes in terms of the quality and diversity of generated results, via comparing to the baseline StarGAN, and is scalable to the multi-font generation via comparing to existing methods for the single-style font generation.

1. Introduction

The generation of Chinese fonts has attracted rising attention within recent years [1, 2, 3, 4], since it is involved in many applications such as the automatic generation of artistic Chinese calligraphy [5], the art font design [6] and personal style generation of Chinese characters [7].

Existing Chinese font generation methods can be generally divided into two categories. The first kind of methods [6, 8] focus on the stroke/radical-level features. These methods mainly exploit the hierarchical representations of strokes/radicals of Chinese characters, and then merge them by some machine learning methods such as support vector machines (SVM) [9], where the hierarchical representations of strokes/radicals are usually yielded manually, and thus this kind of methods are time and effort cost. Moreover, these methods pay much attention to the local representations of the characters, while sometimes overlook the overall style of generated characters. Thus, some postprocessing is usually required to adjust the generated characters.

The second kind of methods [1, 2, 3, 4] are mainly based on the image-level features. The main idea of these methods is to regard the Chinese character generation problem as certain a style transfer learning problem from a source domain to a target domain [10, 11], and then learn the overall style of the input characters via deep generative models, particularly, the
generative adversarial networks (GAN) based models [12]. In light of this idea, the recent work [1] extended the pix2pix model [10] developed for the image-to-image translation to the character-to-character translation of Chinese characters, and then proposed an efficient method called zi2zi based on the paired training samples, of which the collection is generally very cost. In order to break the hurdle brought by the collection of paired training dataset, [2] adapted the cycle GAN (CycleGAN) [11] to the handwritten Chinese character generation (CCG) with unpaired training samples, where their method is called HCCG-CycleGAN. Both [1] and [2] focus on the generation of a single-style Chinese font, while when adapted to the generation of multi-style Chinese fonts, these methods are generally time inefficient with poor scalability.

Figure 1. Generated results of the proposed method on CASIA-HWDB1.1 and three other Chinese character datasets with standard fonts including regular script, middle Song and italics, as well as their associated counterparts in red color. For each style, there are two attributes: character font and character color, where the character font attribute has four attribute values, i.e., \{Regular Script, Middle Song, Italics, Handwritten\}, and the character color attribute is only considered with two attribute values, i.e., \{red, black\}.

There are few works on the generation of multi-style Chinese fonts [4]. The recent paper [4] extended zi2zi from the single-style generation to the multi-style case with the paired training data, of which the collection is very cost. Moreover, it is well known that the GAN-based models usually suffer from the mode collapse issue [12], that is, the model generates the same mode for different inputs. Such phenomenon of GAN will happen more frequently when applied to Chinese font generation, since there are many Chinese characters having the similar strokes.

In this paper, we exploit the popular star generative adversarial networks (StarGAN) [13] to realize the generation of multi-style Chinese fonts with unpaired training dataset. Particularly, in order to alleviate the mode collapse issue of StarGAN, we introduce certain diversity regularizer to the training loss. A series of experiments are conducted on a handwritten Chinese character dataset called CASIA-HWDB1.1 and three standard font datasets of Chinese characters, to demonstrate the effectiveness of the proposed method. The experiment results show that the mode collapse issue of StarGAN when applied to the generation of Chinese characters can be significantly alleviated by the introduced diversity regularizer. Therefore, the diversity and quality of the generated results by StarGAN can be substantially improved, where some generated results of the proposed method are presented in Figure 1. Moreover, the suggested method is much more scalable to the multiple Chinese font generation via comparing to HCCG-CycleGAN [2], an efficient method for a single Chinese font generation.
2. Preliminary Work

In this section, we introduce some related work on the generative adversarial networks (GAN).

A. Generative adversarial networks. GAN [12] has achieved great successes in various applications such as image-to-image translation [11]. GAN commonly includes two competing networks: a generator network and a discriminator network. The discriminator network learns to distinguish between real and fake samples, while the generator network generates fake samples as realistic as possible so that the discriminator can not distinguish these fake samples from real samples. This adversarial nets framework can be formally formulated as the following,

\[
\min_G \max_D \mathbb{E}_{x \sim P_{data}} [\log D(x)] + \mathbb{E}_{z \sim P_z} [\log (1 - D(G(z)))]
\]

where \(P_{data}\) and \(P_z\) are respectively the distributions of data \(x\) and input noise \(z\) for the generator.

B. Conditional GAN (cGAN). Note that GAN is a generative model that learns a mapping from an input noise \(z\) to an output sample \(y = G(z)\) without any auxiliary information. However, in many real-world applications such as image translation [10] and the Chinese font generation considered in this paper, some labels of images or characters are generally specified in advance. Thus, several conditional GANs have been developed in many applications (see, [10, 1]). Mathematically, cGAN is trained to produce an output sample \(y\) from an input noise \(z\) conditioned on certain conditional domain information \(c\), that is, learning a mapping \(G\) such that \(y = G(z, c)\). Thus, the adversarial loss of cGAN should be slightly modified as the following\(^1\),

\[
\mathcal{L}_{adv}(D, G) := \mathbb{E}_{x \sim P_{data}} [\log D(x)] + \mathbb{E}_{z \sim P_z, x \sim P_x} [\log (1 - D(G(z, c)))]
\]

where \(P_x\) represents the distribution of the referred conditional information \(c\).

C. Cycle GAN (CycleGAN). CycleGAN was developed in [11] to tackle the difficulty brought by the collection of paired training data in the training of GAN [12] for the image-to-image translation. It was also extended by [2] to the handwritten Chinese character generation. The main idea of CycleGAN is to preserve key attributes between the input (source) and output (target) domains by utilizing a cycle consistency loss. Let \(x \in \mathcal{X}\) and \(x' \in \mathcal{X}'\) be two style domains conditioned on respectively some conditional information domains \(c \in \mathcal{C}\) and \(c' \in \mathcal{C}'\). In order to realize the bidirectional translation between them, two generators \(G : \mathcal{X} \times \mathcal{C} \rightarrow \mathcal{X}'\) and \(G' : \mathcal{X}' \times \mathcal{C}' \rightarrow \mathcal{X}\) are generally required to be trained. Then the (bi)cycle consistency loss \(\mathcal{L}_{cyc}\) can be defined as the following general form

\[
\mathcal{L}_{cyc}(G, G') = \mathbb{E}_{x \sim P_{x}, c \sim P_{c}, c' \sim P_{c'}}[\|x - G'(G(x, c), c')\|_1] + \mathbb{E}_{x' \sim P_{x'}, c' \sim P_{c'}, c \sim P_{c}}[\|x' - G(G'(x', c'), c)\|_1],
\]

where \(x \in \mathcal{X}\), \(x' \in \mathcal{X}'\), \(c \in \mathcal{C}\), \(c' \in \mathcal{C}'\), and \(P_x\) represents the distribution of the associated domain. \(*\). The generators \(G\) and \(G'\) are trained to make the cycle consistency loss \(\mathcal{L}_{cyc}(G, G')\) small. Intuitively, such a loss is imposed to guarantee that \(G'(G(x, c), c') \approx x\) and \(G(G'(x', c'), c) \approx x'\) for any \(x \in \mathcal{X}\), \(x' \in \mathcal{X}'\) and their associated conditional information \(c, c'\). Particularly, if the bidirectional translation between \(\mathcal{X}\) and \(\mathcal{X}'\) shares the same generator \(G\), and if the one-side direction cycle consistency loss (say, from the source domain \(\mathcal{X}\) to the target domain \(\mathcal{X}'\)) is only considered, then the cycle consistency loss yields the following reconstruction loss \(\mathcal{L}_{rec}\).

\[
\mathcal{L}_{rec}(G) = \mathbb{E}_{x \sim P_{x}, c \sim P_{c}, c' \sim P_{c'}}[\|x - G(G(x, c), c')\|_1].
\]

We will omit the distributions \(P_x\) in the expectation if there is no confusion from the context.

3. Diversity Regularized StarGAN

In this section, we introduce the proposed diversity regularized StarGAN (called D-StarGAN for short) for the multi-style Chinese font generation.

\(^1\) Note that the conditional information on the generator is only considered in (2), while if the discriminator \(D\) is also imposed on the conditional information, the adversarial loss can be further modified in a similar manner as considered in [10].
3.1. Multi-style Chinese Font Generation

The multi-style Chinese font generation problem can be described as follows: translating Chinese characters of a given style into Chinese characters of multiple styles. In this paper, we mainly focus on two attributes, font and color. Specifically, in our later experiments, the font attribute mainly takes four values with \{Regular Script, Middle Song, Italics, Handwritten\}, and the color attribute mainly takes two values with \{red, black\}. For each attribute, we use a one-hot vector representation. Thus, the label of each style \(c\) is represented as a contaminated vector of these two one-hot vectors for both attributes. For example, the label \(c\) of the style Handwritten font with black color is represented as \(0 0 0 1 0 1\), where the first four components \(0 0 0 1\) and the last two components \(0 1\) are respectively the one-hot vector representations of Handwritten font and black color. Let \(X\) be an input domain of Chinese characters, and \(C := \{c_i\}_{i=1}^n\) be a set of specified styles of Chinese characters (generally called conditional domain). Then the main purpose of multi-style Chinese font generation is to learn a mapping \(G\) from the input domain and the conditional style set to the output domain such that \(y = G(x, c), \forall x \in X, c \in C\).

3.2. StarGAN for Multi-style Chinese Font Generation

To efficiently solve the above multi-style Chinese font generation problem, we use the network architecture of StarGAN [13] recently developed for the multi-domain image translation. When adapted to the multi-style Chinese font generation, the working flow of StarGAN is described in Figure 2. From Figure 2, StarGAN consists of only one discriminator \(D\) and one generator \(G\). The tasks of the generator \(G\) are two folds: i) learning to generate realistic-like fake characters from the input characters and the target style labels to fool the discriminator, and ii) reconstructing the original input characters as accurately as possible from the fake characters generated by itself in the translated style domain and the original style labels. The discriminator \(D\) also undertakes two tasks: i) distinguishing the fake characters from the real characters, and ii) classifying the real and fake characters into their corresponding style labels. More specifically, the discriminator \(D\) of StarGAN is designed to produce probability distributions over both the input (also called source) domain and the style set, that is, \(D : x \rightarrow (D_{src}(x), D_{cls}(c|x))\) \(^2\), where \(D_{src}(x)\) represents the probability distribution over the source domain for a given input \(x\) of \(D\), and \(D_{cls}(c|x)\) represents the classification probability that \(x\) is classified as the class with the

\(^2\) The subscripts \(src\) and \(cls\) are respectively the abbreviations of these two vocabularies source and classification.
specified style label $c$. In the following, we establish the training model of StarGAN in detail.

A. Adversarial loss. The first part of the training loss of StarGAN is the following adversarial loss,

$$
L_{adv}(D, G) = E_x[\log D_{src}(x)] + E_{x,c}[\log(1 - D_{src}(G(x, c)))] ,
$$

where $G$ generates the fake character $G(x, c)$ conditional over both the input character $x$ and the target style label $c$, while $D$ attempts to distinguish the fake characters and the real characters.

B. Reconstruction loss. As described before, the generator of the StarGAN is designed to not only generate realistic-like fake characters but also reconstruct the original characters conditional on the generated fake characters in the target (or output) style domain and the associated style labels of the original characters in the source domain. Thus, according to (3), the following reconstruction loss should be included in the training loss of StarGAN, that is,

$$
L_{rec}(G) = E_{x,c,c'}[\|x - G(G(x, c), c')\|_1],
$$

where $c'$ and $c$ are respectively the style labels of source and target domains, and $\| \cdot \|_1$ is the $\ell_1$ norm. The reconstruction loss is utilized to make $G$ be revertible such that a single generator can realize the bidirectional translation between two style domains, instead of two generators used in CycleGAN [11, 2] as shown in Section 2 C.

C. Domain classification loss. Besides the adversarial loss and reconstruction loss, certain domain classification loss introduced by the auxiliary classifier to control the multi-style domains should be also considered. Specifically, given an input character $x$ in some source style domain with the style label $c'$, let $G(x, c)$ be an output character of the generator $G$ conditional on the style label $c$ of the target domain. For these input characters and their associated generated characters, $D$ is desired to classify them with the correct style labels as accurately as possible. Thus, the domain classification loss for the input character $x$ with the source style label $c'$ and the generated character $G(x, c)$ with the target style label $c'$ can be defined as

$$
L_{cls}^c(D) = E_{x,c'}[\log D_{cls}(c'|x)], \quad L_{cls}^f(G) = E_{x,c}[\log D_{cls}(c|G(x, c))].
$$

D. Total loss for StarGAN. According to the above defined loss functions (4)-(6), the total training loss of StarGAN for multi-style Chinese font generation is summarized as follows:

$$
L_{stargan}(D, G) = L_{adv}(D, G) + \lambda_{rec}L_{rec}(G) + \lambda_{cls}(L_{cls}^c(D) + L_{cls}^f(G)),
$$

where $\lambda_{rec}$ and $\lambda_{cls}$ are two regularization parameters. Then $D$ attempts to maximize $L_{stargan}$ while $G$ tries to minimize it, that is, $\min_G \max_D L_{stargan}(D, G)$.

3.3. Diversity Regularization for StarGAN Training

It is known that GAN suffers from the mode collapse issue during the training [12]. When mode collapse occurs, several different modes will merge into one such that fewer mode will be generated. Thus, the diversity and quality of generated results will be significantly degraded. When adapted to the multi-style Chinese font generation, due to the similar shape, the mode collapse issue occur more frequently in the training of StarGAN, as shown in Figure 3.

Figure 3. Examples of very similar Chinese characters with the associated pinyin presented in the brackets.
To alleviate this issue, we introduce a diversity regularizer in the training of StarGAN. The intuition behind our idea lies in that given two similar input characters \( x, x' \) in the source domain, the generator \( G \) is desired to preserve and even amplify the difference of \( x \) and \( x' \) such that the associated generated characters \( G(x, c) \) and \( G(x', c) \) in the target domain with the style label \( c \) can be well distinguished. Mathematically, we use the ratio between \( \| G(x, c) - G(x', c) \|_1 \) and \( \| x - x' \|_1 \) (where \( \| \cdot \|_1 \) represents the \( \ell_1 \) norm), i.e.,

\[
L_{\text{div}}(G) = \mathbb{E}_{x, x', c} \left[ \frac{\| G(x, c) - G(x', c) \|_1}{\| x - x' \|_1} \right],
\]

where the diversity regularizer \( L_{\text{div}}(G) \) is taken as the expectation of the ratio over the input domain and the set of style labels. The \( \ell_1 \) norm is mainly due to the sparsity of the difference of two similar characters. In order to improve the diversity and also alleviate the mode collapse issue, \( G \) is trained via maximizing \( L_{\text{div}}(G) \). As a consequence, the training loss for our proposed Diversity Regularized StarGAN (called D-StarGAN for short) can be described as follows,

\[
\min_G \max_D L_{\text{total}}(D, G) := L_{\text{stargan}}(D, G) - \lambda_{\text{div}} L_{\text{div}}(G).
\]

where \( \lambda_{\text{div}} \) is some regularization parameter, set empirically as 1 in the experiments.

**Remark.** Note that a similar diversity regularization of (8) was suggested in [14] for the image style translation. However, in this paper, we define the diversity regularization directly on the pixel level, while the diversity regularization in [14] is defined on some latent space.

To improve the stability of GAN training, the Wasserstein GAN (WGAN) [15] was introduced as an alternative, the total loss \( L_{\text{total}}(D, G) \) of D-StarGAN becomes

\[
L_{\text{total}}(D, G) := L_{\text{wgan}}(D, G) + \lambda_{\text{rec}} L_{\text{rec}}(G) + \lambda_{\text{cls}}(L_{\text{cls}}(D) + L_{\text{cls}}^f(G)) - \lambda_{\text{div}} L_{\text{div}}(G).
\]

### 4. Numerical Experiments

In this section, we conducted a series of experiments to demonstrate the efficiency and effectiveness of the proposed D-StarGAN. All experiments were carried out in Pytorch environment running Linux, AMD(R) Ryzen 7 2700x eight-core processor \( \times 16 \) CPU, GeForce RTX 2080 GPU. The codes are available at [https://github.com/JinshanZeng/MS-StarGAN](https://github.com/JinshanZeng/MS-StarGAN).

#### 4.1. Experimental Settings

The experimental settings are described as follows.

**A. Datasets and Evaluation metrics.** The dataset used in this paper is composed of 8 sub-datasets of different styles, that is, one benchmark dataset for the handwritten Chinese characters in black color called CASIA-HWDB1.\(^3\) and three benchmark datasets for the Chinese characters with standard fonts \{Regular script, Middle Song, Italics\} in black color, as well as their counterparts in red color. Our dataset includes 30K training samples and 2K test samples in total. Each sample is stored in the image format and resized to \( 128 \times 128 \times 3 \).

The quality of generated characters was evaluated by the following two measures: content accuracy and style discrepancy as commonly used in the literature [2]. More specifically, we used a pre-trained HCCG-GoogLeNet [16] model to evaluate content accuracy of generated Chinese characters. Intuitively, if the pre-trained model can correctly classify the characters that we generate, it means that the quality of our generated sample is good. Thus, in terms of the content accuracy, higher accuracy implies better performance. In order to evaluate the style discrepancy, we exploited the well-known Frechet Inception Distance (FID) [17], which is

\(^3\) [http://www.nlpr.ia.ac.cn/databases/handwriting/Home.html](http://www.nlpr.ia.ac.cn/databases/handwriting/Home.html)
a sample distance measure. FID is the distance between the true and generated distributions. Thus, smaller FID generally implies that the generated samples are closer to the real samples, and hence better performance of the considered method.

**B. Baselines.** In order to show the effectiveness and efficiency of the proposed D-StarGAN, we took the original StarGAN adapted to multi-style Chinese font generation and HCCG-CycleGAN [2] for the single Chinese font generation as the baselines. Moreover, we also considered the following models as baselines in different experiments:

(i) **StarGAN-SN:** StarGAN incorporated with spectral normalization for both generator and discriminator networks,

(ii) **D-StarGAN-20:** D-StarGAN with 20-layer generator (that is, adding one more residual network module incorporated with spectral normalization to D-StarGAN).

4.2. Experimental Results

In the following, we describe the main results of the conducted experiments.

**A. Improving stability via spectral normalization.** A minor adaptation of the networks used in this paper is that the spectral normalization [18] is incorporated into both generator and discriminator networks of the original StarGAN [13]. Under the same setting, we trained StarGAN and StarGAN-SN over the built dataset. The curves of total training loss values for both StarGAN and StarGAN-SN are shown in Figure 4. From Figure 4, it can be observed that the training stability of StarGAN is improved by the use of spectral normalization, as also observed in the literature [18]. Such an improvement on the training stability may also lead to that on the generation quality of the Chinese characters, as shown in Figure 5. It can be observed that the generated characters by StarGAN-SN are better than those by StarGAN, particularly, the characters in the handwritten font style.

**B. Alleviating mode collapse via diversity regularization.** In this experiment, we show the effect of the introduced diversity regularization when comparing with StarGAN-SN, the counterpart of the proposed D-StarGAN without imposing the diversity regularization. Some generated results of D-StarGAN and StarGAN-SN with very similar Chinese characters (shown in Figure 3). We only present the generated results in the style of handwritten font with black

---

**Figure 4.** Total training loss of both StarGAN and StarGAN-SN.

**Figure 5.** Some generated results of StarGAN and StarGAN-SN. The first and second rows of results are generated by StarGAN and StarGAN-SN, respectively.
color in Figure 3, since the mode collapse phenomenon is frequently observed for the generation of such style font. From Figure 6, it can be observed that StarGAN-SN suffers from the mode collapse issue, as shown in the second row of Figure 6. However, such a mode collapse issue is alleviated to some extent by the introduced of diversity regularization, as shown by the third row of Figure 6. This verifies the effectiveness of the introduced diversity regularization in our proposed StarGAN training. Besides the comparison between D-StarGAN and StarGAN-SN over the similar Chinese characters, we also compare their performance for more general Chinese characters, as shown in Figure 7. From Figure 7, the quality of the generated characters can be improved by introducing the diversity regularization, and the improvement is in particular significant when applied to generate the handwritten Chinese characters. All these results demonstrate the effectiveness and to some extent, the necessity of the diversity regularization.

Figure 6. Comparison on the performance of StarGAN-SN and D-StarGAN with very similar Chinese characters as the test inputs.

Figure 7. Comparison on the performance of StarGAN-SN and D-StarGAN with some general Chinese characters as the test inputs.

C. Scalability of D-StarGAN. In this experiment, we aim to show the scalability of the proposed D-StarGAN to multi-style Chinese font generation via comparing with HCCG-CycleGAN [2] developed for a single-style Chinese font generation, that is, the handwritten Chinese font generation. For this purpose, we trained D-StarGAN for several multi-style font generation tasks with the number of styles varying from 2 to 6, and then recorded the training time, as well as the content accuracy and FID for each case. Besides HCCG-CycleGAN, we also considered D-StarGAN-20 by adding one more residual block than the suggested D-StarGAN, to show the potential effect of the depth of neural networks. The curves are shown in Figures 8.

From Figure 8(a), as the style number increases from 2 to 6, the training time of D-StarGAN increases very slowly, as shown in the reddish orange curve, while the training time of HCCG-CycleGAN is 10.6 hours only for the character translation between two styles. In order to realize the multi-style Chinese font generation with \( n \) styles, the total training time of HCCG-CycleGAN is estimated to be \( n \times (n - 1) \) times of 10.6 hours (say, 318 hours in total for only 6 styles), which is much more than the proposed D-StarGAN. Thus, in terms of training time, D-StarGAN is well scalable to the multi-style Chinese font generation. Moreover, the blue curve is the training time increases as the depth of generator network increases. Furthermore, by Figure 8(b,c) both content accuracy and FID of D-StarGAN are stable with respect to the style number, shown by the reddish origin color curves, although the trends of both content accuracy and FID value of D-StarGAN slightly get worse as the style number increases, mainly due to the increase of the training difficulty as the style number increasing. When compared to D-StarGAN-20 (i.e.,
D-StarGAN with the 20-layer generator network), the suggested D-StarGAN (with an 18-layer generator network) is slightly worse in terms of content accuracy, but slightly better in terms of FID, reflecting the better diversity of characters generated by D-StarGAN. Particularly, in the viewpoint of vision, the suggested D-StarGAN usually generates characters with slightly higher quality than D-StarGAN-20, as shown in Figure 9.

![Figure 8](image1.png)

**Figure 8.** Content accuracy and FID of D-StarGAN with respect to the number of styles.

![Figure 9](image2.png)

**Figure 9.** Comparison on the generated Chinese characters by D-StarGAN and D-StarGAN-20. It can be observed that the quality of the characters generated by D-StarGAN is slightly better than that of characters generated by D-StarGAN-20.

5. Conclusion
This paper proposes an efficient deep generative method for the multi-style Chinese font generation problem. Our proposed method is built upon the well-known StarGAN. In order to tackle the mode collapse issue of StarGAN, we introduce a diversity regularizer incorporated into the training of StarGAN. Moreover, the spectral normalization is also exploited to improve the stability of StarGAN training. A series of experiments are conducted on a handwritten Chinese character dataset called CASIA-HWDB1.1 and three benchmark datasets for the Chinese characters with standard fonts, to demonstrate the feasibility and effectiveness of our proposed method. Motivated by [19], one future direction is how to exploit the stroke information of Chinese characters to improve the generation quality of the proposed method.

Acknowledgment
This work is supported in part by the National Natural Science Foundation of China [Grant Nos. 61977038, 61876074], and by the Thousand Talents Plan of Jiangxi Province. Part of J. Zengs work was done when he visited at City University of Hong Kong. The authors would like to thank Prof. Ding-Xuan Zhou and Dr. Wenyan Xiao for their great help in the discussion and revision of this paper.

References
[1] Yuchen Tian, Zi2zi: Master Chinese calligraphy with conditional adversarial networks. https://github.com/kaonashi-tyc/zi2zi. 2017.
[2] Bo Chang, Qiong Zhang, Shenyi Pan and Lili Meng, Generating handwritten Chinese characters using CycleGAN. Proc. Winter Conference on Applications of Computer Vision (WACV’18), Lake Tahoe, USA, 2018, 1–9.

[3] Weirui Kong and Bicheng Xu, Handwritten Chinese character generation via conditional neural generative models. Proc. the 31st Conference on Neural Information Processing Systems (NeurIPS’17), Long Beach, CA, 2017, 1–7.

[4] Jiefu Chen, Yanli Ji, Hua Chen and Xing Xu, Learning one-to-many stylised Chinese character transformation and generation by generative adversarial networks. IET Image Processing, 2019, (13)2680–2686.

[5] Boheng Zhao, Jianhua Tao, Minghao Yang, Zhengkun Tian, Cunhang Fan and Ye Bai, Deep Imitator: Handwriting calligraphy imitation via deep attention networks. Pattern Recognition, 2020, (104)11–14.

[6] Jeng-Wei Lin, Chih-Yin Wang, Chao-Lung Ting and Ray-I Chang, Font generation of personal handwritten Chinese characters. Proc. International Conference on Graphic and Image Processing (ICGIP’14), Beijing, China, 2014, 1–6.

[7] Peng Liu, Songhua Xu and Shujin Lin, Automatic generation of personalized Chinese handwriting characters. Proc. the 4th International Conference on Digital Home (ICDH’12), Guangzhou, China, 2012, 421–425.

[8] Hiroshi Nagahashi and Mikio Nakatsuyama, A pattern description and generation method of structural characters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986, (Volume: PAMI-8, Issue: 1) 112–118.

[9] Nello Cristianini and John Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel Based Learning Methods. Cambridge University Press. 2000.

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou and Alexei A. Efros, Image-to-Image translation with conditional adversarial networks. Proc. the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR’17), Honolulu, Hawaii, 2017, 2223–2232.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza and Bing Xu, Generative adversarial nets. Proc. the 28th Conference on Neural Information Processing Systems (NeurIPS’14), Montreal, Canada, 2014, 1–9.

[12] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim and Jaegul Choo, StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. Proc. the 31st IEEE Conference on Computer Vision and Pattern Recognition (CVPR’18), Salt Lake City, 2018, 8789–8797.

[13] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma and Ming-Hsuan Yang, Mode seeking generative adversarial networks for diverse image synthesis. Proc. the 32nd IEEE Conference on Computer Vision and Pattern Recognition (CVPR’19), Long Beach, CA, 2019, 1429–1437.

[14] Martin Arjovsky, Soumith Chintala and Leon Bottou, Wasserstein generative adversarial networks. Proc. the 34th International Conference on Machine Learning (ICML’17), Sydney, Australia, 2017, 214–223.

[15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich, Going Deeper with Convolutions. Proc. the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR’14), Columbus, Ohio, 2014, 1–9.

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler and Sepp Hochreiter, GANs trained by a two time-scale update rule converge to a local Nash equilibrium. Proc. the 31st Conference on Neural Information Processing Systems (NeurIPS’17), Long Beach, CA, 2017, 1–12.

[17] Takeru Miyato, Toshiki Kataoka, Masanori Koyama and Yuichi Yoshida, Spectral normalization for generative adversarial networks. Proc. International Conference on Learning Representations (ICLR’18), Vancouver, Canada, 2018, 1–26.

[18] Jinshan Zeng, Qi Chen, Yunxin Liu, Minwen Wang and Yuan Yao, StrokeGAN: Reducing mode collapse in Chinese font generation via stroke encoding. Proc. the 35th AAAI Conference on Artificial Intelligence (AAAI’2021), 2021, 1–9.