The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)

HiGAN: Handwriting Imitation Conditioned on Arbitrary-Length Texts and Disentangled Styles

Ji Gan, Weiqiang Wang *

School of Computer Science and Technology, University of Chinese Academy of Sciences
ganji15@mails.ucas.ac.cn, wqwang@ucas.ac.cn

Abstract

Given limited handwriting scripts, humans can easily visualize (or imagine) what the handwritten words/texts would look like with other arbitrary textual contents. Moreover, a person also is able to imitate the handwriting styles of provided reference samples. Humans can do such hallucinations, perhaps because they can learn to disentangle the calligraphic styles and textual contents from given handwriting scripts. However, computers cannot study to do such flexible handwriting imitation with existing techniques. In this paper, we propose a novel handwriting imitation generative adversarial network (HiGAN) to mimic such hallucinations. Specifically, HiGAN can generate variable-length handwritten words/texts conditioned on arbitrary textual contents, which are unconstrained to any predefined corpus or out-of-vocabulary words. Moreover, HiGAN can flexibly control the handwriting styles of synthetic images by disentangling calligraphic styles from the reference samples. Experiments on handwriting benchmarks validate our superiority in terms of visual quality and scalability when comparing to the state-of-the-art methods for handwritten word/text synthesis. The code and pre-trained models can be found at https://github.com/ganji15/HiGAN.

Introduction

Handwriting imitation aims at (1) synthesizing diverse, realistic handwriting images conditioned on arbitrary textual contents and (2) imitating the calligraphic styles (e.g., the text skew, slant, roundness, ligature, and stroke-width) of reference handwriting samples. As shown in Fig. 1, humans can quickly learn such handwriting imitation with the ability of hallucination. Explicitly speaking, after learning from limited handwriting documentations, a person can visualize (or imagine) what the handwritten words/texts would look like conditioned on other arbitrary textual contents. Furthermore, providing a reference handwriting sample, a person can also hallucinate new handwriting images of similar calligraphic styles but with different textual contents. Humans can do such hallucinations, perhaps because they can learn to disentangle the calligraphic styles and textual contents from provided handwriting samples. If we can teach computers to mimic this process, they possibly are capable of imitating realistic handwriting as well as humans.

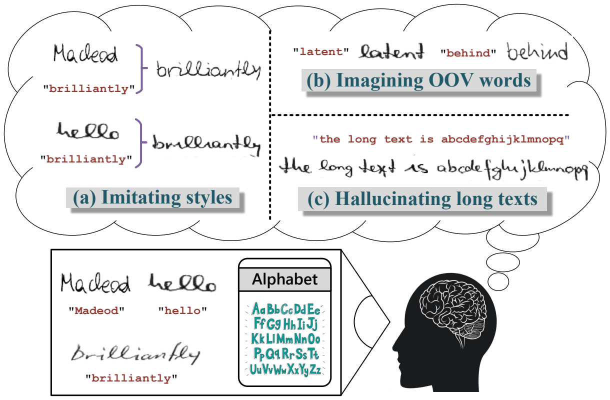


Figure 1: Humans can easily learn the handwriting imitation with the ability of hallucination.

With the invention of variational auto-encoders (VAEs) (Kingma and Welling 2013) and generative adversarial networks (GANs) (Ian Goodfellow 2014), it has witnessed significant progress in the field of image synthesis in recent years. Combining conceptions of VAEs and GANs, computers nowadays can not only synthesize realistic images but also perform image-to-image translation between different visual domains (Zhu et al. 2017; Andrew Brock and Simonyan 2019; Yunjey Choi and Choo 2018; Yunjey Choi 2020; Hsin-Ying Lee and Yang 2018). However, a significant observation is that VAEs/GANs for handwriting synthesis mainly focus on generating fixed-sized images of isolated digits/characters. On the contrary, very few works have explored to synthesize handwritten words/texts with GANs. Particularly, we demonstrate that handwritten word/text synthesis is much more challenging than isolated character generation, since it faces the following three difficulties:

- **Variable-Sized Images.** The model must be capable of synthesizing variable-sized images, since handwriting images of variable-length texts should have different sizes (e.g., text images typically are longer than word images).
- **Arbitrary Words/Texts.** Given the language alphabet, the model should generate readable handwriting images conditioned on arbitrary texts that unconstrained to any predefined corpus or out-of-vocabulary (OOV) words.

*Corresponding Author: Weiqiang Wang

Features	**ScrabbleGAN** **Fogel (2020)**	**GANwriting** **Kang (2020)**	**HiGAN** **Ours**
Random Styles	√	×	√
Desired Styles	×	√	√
Short Words	√	√	√
Long Texts	√	×	√

Table 1: Feature-by-feature comparison of GANs for handwritten word/text generation. "Random Styles" denotes whether models can generate images by randomly sampling styles from a prior distribution; "Desired Styles" indicates whether models can imitate handwriting styles of reference images; and "Short Words" & "Long Texts" denotes whether models can synthesize short words or long texts respectively.

- **Style Imitation.** The model should learn to disentangle calligraphic styles from variable-sized reference images, and then imitate the extracted styles for synthetic images.

Overall, those characteristics make handwritten text generation different from conventional image synthesis.

Recently, several efforts have been made to synthesize handwritten word/text images with GANs; nevertheless, none of them has successfully overcome the aforementioned difficulties all at once. Specifically, the first attempt (Eloi Alonso and Messina 2019) is to generate fixed-sized handwritten word images conditioned on the learned embeddings of entire words. Hence, their method cannot generate either variable-length images or high-quality handwritten OOV words. One improved work is ScrabbleGAN (Sharon Fogel and Litman 2020), which can synthesize long handwritten texts by concatenating all letter-tokens together. However, both methods cannot learn to imitate the calligraphic styles of reference samples. Another improved work is GANwriting (Lei Kang 2020), which can generate handwritten words conditioned on calligraphic style features. Still, GANwriting cannot generate long handwritten words/texts (with more than ten letters) due to its flaws in architecture design. What is more, GANwriting requires multiple reference images to extract reliable styles for high-quality synthesis, thus exhibiting low visual quality in inference if only one reference sample is provided at test time. Summarizing, the state-of-the-art GANs have not entirely solved the handwritten word/text synthesis yet.

In this paper, we propose a novel handwriting imitation GAN (HiGAN). The proposed method can generate variable-length handwritten words/texts conditioned on arbitrary textual contents, which are unconstrained to any predefined corpus or out-of-vocabulary words. Moreover, HiGAN can flexibly control the handwriting styles of synthetic images, which is achieved by disentangling the calligraphic styles from reference samples. Specifically, Table 1 shows the feature-by-feature comparison between HiGAN and the competing GANs. Our contribution is twofold:

- We propose a novel HiGAN for handwriting imitation, which can (1) generate variable-sized handwriting images condition on arbitrary texts and (2) imitate calligraphic styles that disentangled from the reference samples.
- Experiments on handwriting benchmarks validate our superiority in terms of visual quality and scalability compared with the state-of-the-art GANs for handwritten word/text synthesis.

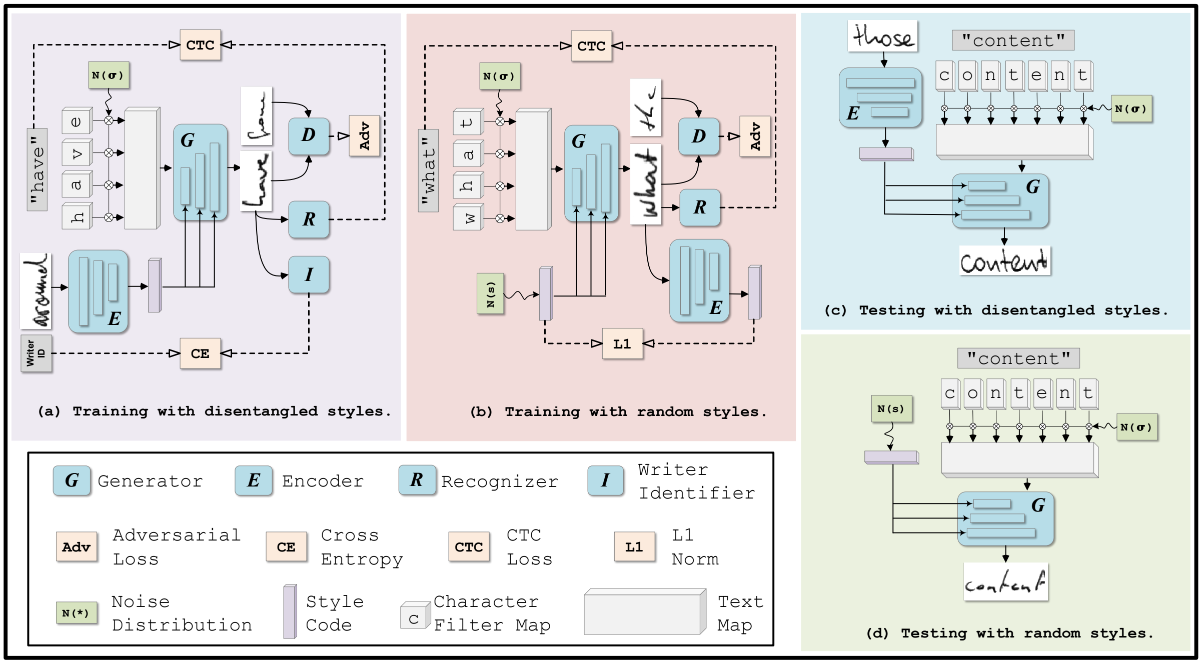
Related Work

Teaching machines to synthesize realistic handwriting is an interesting topic. Although it has witnessed many significant achievements in recent years, handwriting imitation remains challenging for artificial intelligence. Traditional handwriting synthesis methods (Tom S. F. Haines and Brostow 2016; Lin and Wan 2007; Achint Oommen Thomas and Govindaraju 2009) typically involve expensive manual interventions, which require the strong domain-specific knowledge aimed at clipping isolated characters, rendering backgrounds and ligatures among strokes.

With the great success of deep learning in machine learning, artificial neural networks have gradually been introduced to handwriting synthesis. The first relevant work is that Graves (2013) proposed to synthesize online handwriting trajectories with recurrent neural networks (RNNs). However, such recurrent-based model not only suffers from the difficulty of learning long-range dependencies, but it also is time-consuming due to its computation mechanism. More lethally, it may be challenging to collect massive trajectories in practical scenarios, since the unique equipment (e.g., digital pens) is required to record the sequential points. In contrast, it is much easier to obtain a mass of handwriting images in our real lives and the internet. Thus, our primary objective in this paper is to synthesize handwriting images.

With the advances of generative models, handwritten character synthesis has been fully explored in recent years. Specifically, both GANs (Ian Goodfellow 2014) and VAEs (Kingma and Welling 2013) were proposed to synthesize realistic handwritten digits. Moreover, Mirza *et al.* (2014) further proposed conditional GANs (cGANs) for guiding machines to generate digits conditioned on class labels. Furthermore, Radford *et al.* (2013) proposed deep convolutional GANs (DCGANs) to synthesize more realistic images by redesigning the structures of GANs with deep convolutional networks. Despite digits, Chang *et al.* (2018) also proposed to utilize CycleGANs (Jun-Yan Zhu 2017) for synthesizing isolated Chinese characters with thousands of categories.

Heretofore, most existing GANs focus on fixed-sized isolated character synthesis, yet only a few efforts have been made for handwritten word/text generation. Specifically, Alonso *et al.* (2019) first proposed to generate fixed-sized handwritten word images using GANs while producing poor visual quality. Furthermore, Fogel *et al.* (2020) proposed ScrabbleGAN to synthesize variable-length handwritten texts with randomly sampled styles. However, a significant shortcoming is that they cannot imitate the calligraphic styles of reference images. More recently, Kang *et al.* (2020) proposed GANwriting, which can generate short handwritten words conditioned on provided style images. However, their method is limited to synthesizing short handwritten words (with less than ten letters), and thus it cannot generate long handwritten texts. Moreover, it exhibits low vi-



(a) Training with disentangled styles.

(b) Training with random styles.

(c) Testing with disentangled styles.

(d) Testing with random styles.

Figure 2: The overview of HiGAN. During training, the model learns to (a) disentangle calligraphic styles from real images and (b) generate fake images that are indistinguishable from real ones. At test time, the model can either (c) imitate the calligraphic styles that disentangled from reference samples or (d) just randomly sample styles from a prior distribution to generate different handwriting images. In addition, each module shares its parameters at different training stages of (a) and (b).

sual quality when providing only one reference image at test time. Summarizing, the state-of-arts GANs are still unable to synthesize realistic variable-sized handwriting images with controllable handwriting styles. In contrast, our HiGAN is capable of generating diverse handwriting words/texts conditioned on arbitrary-length texts and disentangled styles.

Handwriting Imitation GAN

Given a set of handwriting images $\mathcal{X}$ and their text labels $\mathcal{Y}$, our goal is to train a generator G that can synthesize diverse, realistic handwriting images conditioned on the arbitrary-length text $\mathbf{y} = [y_1, \cdots, y_L]$ (with a length of L) and any style feature s. Notably, the style s can be either (1) sampled from a prior distribution $\mathcal{N}(0, 1)$ or (2) disentangled from a reference image $x \in \mathcal{X}$ with a pre-trained style encoder E (i.e., $s = E(x)$). Specifically, Fig. 2 illustrates the overview of HiGAN. Next, we will describe the framework of HiGAN and its training objectives in the following subsections.

Framework of HiGAN

In this subsection, we will detail the framework of the proposed HiGAN, which consists of the following components:

Generator. Since we require the model to generate handwriting images conditioned on arbitrary-length text $\mathbf{y}$, the generator G must be capable of producing variable-length images (rather than that with fixed sizes). For handwriting, one strong assumption is that humans typically finish a text with writing letters sequentially and individually, while the corresponding ligatures mostly depend on the adjacent characters. Hence, the generator G is designed to mimic this handwriting process with the following strategies:

1. **Text-Map Generation.** Let $\mathcal{F} = \{f_c | c \in \mathcal{A}\}$ be a set of character filter maps, where $\mathcal{A}$ is the alphabet and f_c is the filter map of character c. Furthermore, each filter map f_c is modulated with a consistent but randomized noise σ to introduce subtle distortions. As a result, a given text $\mathbf{y} = [y_1, \cdots, y_L]$ can be embedded into multiple filter maps as $\mathcal{F}(\mathbf{y}, \sigma) = [f_{y_1} \otimes \sigma, \cdots, f_{y_L} \otimes \sigma]$. Finally, those filter maps are concatenated horizontally into a wide one, regarded as a style-invariant text-map $\mathcal{M}$.

2. **Style Rendering.** A fully convolutional network is further utilized to upsample the text-map $\mathcal{M}$ and simultaneously render calligraphic styles. Specifically, conditional batch normalization (CBN) (Harm de Vries and Courville 2017) is employed to inject the style feature s into G, explicitly affecting the handwriting styles of synthetic images (such as the text slant, skew, stroke width, and character shape). Since CBN is applied globally over the whole text-map, all letters are guaranteed to keep the consistent styles. Moreover, due to the merits of convolutions, the generator can automatically learn overlaps between adjacent letters and draw natural ligatures if necessary.

This eventually leads to the generator G being able to synthesize variable-length handwriting images conditioned on the arbitrary text $\mathbf{y}$ with a controllable style s.

Discriminator. The discriminator D mainly learns a binary classification determining whether an image x is a real image from the training set or a fake image generated by G.

Style Encoder. The style encoder E can disentangle the handwriting style s from a reference image x but without explicitly accessing the corresponding writer identity.

Writer Identifier. The writer identifier I is utilized to distinguish which writer a handwriting image x belongs to. Notably, the identifier I can only classify the handwriting images of seen writers from the training set, while it cannot identify the handwriting of unseen writers at test time. Therefore, the identifier I is only employed at training time, which aims at guiding the encoder E to cluster embeddings for the images of similar styles but diversify embeddings for the images of distinct styles (e.g., the samples from different writers). Hence, the identifier I is discarded at test time.

Recognizer. Given a handwriting sample x, the recognizer R is supposed to predict its text labels $\mathbf{y}$ correctly. Notably, the recognizer R is first trained with the annotated data $\{\mathcal{X}, \mathcal{Y}\}$, and then is utilized to guide the generator G to synthesize readable handwriting with any desired textual content $\mathbf{y}$. Since the training corpus typically has limited semantic knowledge, the recognizer R cannot generalize to an open language domain. As a result, it may be difficult for R to correctly recognize OOV words, while which is not desired for handwriting synthesis. One feasible solution is to remove the recurrent neural network, thus leaving R with a fully convolutional structure. This eventually prevents R from learning an implicit language model constrained on the training corpus, which may benefit OOV word generation.

Training Objectives

To train HiGAN, it requires a set of handwriting images $\mathcal{X}$, their text labels $\mathcal{Y}$ and the corresponding writer identities $\mathcal{W}$. Moreover, a large open corpus $\mathcal{C}$ is employed to yield arbitrary texts for generating fake images at training time, where $\mathcal{Y} \subset \mathcal{C}$. As shown in Fig. 2 (a) & (b), we train our framework with the following objectives:

Adversarial Loss. Given an arbitrary text $\tilde{\mathbf{y}} \in \mathcal{C}$ and a style feature s that sampled from a prior normal distribution $\mathcal{N}(0,1)$, the generator G learns to synthesize a fake image $G(\tilde{\mathbf{y}}, s)$ that is indistinguishable (by the discriminator D) from the real image $x \in \mathcal{X}$ via the adversarial loss, i.e.,

$$\mathcal{L}_{adv_1} = \mathbb{E}_x[\log D(x)] + \mathbb{E}_{\tilde{\mathbf{y}},s}[\log(1 - D(G(\tilde{\mathbf{y}}, s)))]. \quad (1)$$

Furthermore, given the real image x as a reference, the generator should also synthesize a realistic image conditioned on the disentangled style $E(x)$ as

$$\mathcal{L}_{adv_2} = \mathbb{E}_x[\log D(x)] + \mathbb{E}_{\tilde{\mathbf{y}},x}[\log(1 - D(G(\tilde{\mathbf{y}}, E(x))))]. \quad (2)$$

Therefore, the overall adversarial loss during training is the sum of the two (i.e., $\mathcal{L}_{adv_1}$ and $\mathcal{L}_{adv_2}$) as

$$\mathcal{L}_{adv} = \mathcal{L}_{adv_1} + \mathcal{L}_{adv_2}. \quad (3)$$

This adversarial loss can guarantee the generator G to synthesize realistic handwriting images.

Text Recognition Loss. Despite visual verisimilitude, the generator G should also constrain the synthetic images to preserve the desired textual contents $\mathbf{y}$. To this end, the recognizer R first is optimized by minimizing the connectionist temporal classification (CTC) (Alex Graves and Gomez 2006) loss for each ground-truth pair $\{x \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}\}$ from the training set, i.e.,

$$\mathcal{L}_{ctc}^{D} = \mathbb{E}_{x,\mathbf{y}}[-\mathbf{y} \log R(x)], \quad (4)$$

when maximizing the adversarial loss. This loss ensures that the recognizer R can correctly predict text labels of the given handwriting image x. Therefore, the trained R can further guide G to synthesize the readable handwriting image $G(\tilde{\mathbf{y}}, s)$ that retains the desired textual contents $\tilde{\mathbf{y}} \in \mathcal{C}$ as

$$\mathcal{L}_{ctc}^{G} = \mathbb{E}_{\tilde{\mathbf{y}},s}[-\tilde{\mathbf{y}} \log R(G(\tilde{\mathbf{y}}, s))], \quad (5)$$

where $\tilde{\mathbf{y}}$ is sampled from the open corpus $\mathcal{C}$, and the parameters of R keep fixed when minimizing the adversarial loss.

Style Disentangling. The primary objective of HiGAN is learning to exactly disentangle calligraphic styles from the reference handwriting images. To this end, we first employ the latent style reconstruction loss to HiGAN as

$$\mathcal{L}_{sty} = \mathbb{E}_{\tilde{\mathbf{y}},s}[||s - E(G(\tilde{\mathbf{y}}, s))||_1], \quad (6)$$

i.e., $\mathcal{L}_{sty}$ enforces the model to reconstruct the style s of any synthetic image $G(\tilde{\mathbf{y}}, s)$ with the help of the encoder E, thus guaranteeing that the style s can explicitly affect handwriting styles of synthetic images.

Furthermore, we also optimize the writer identifier I by minimizing the cross-entropy loss for each ground-truth pair $\{x, w\}$ of the training set (where $w \in \mathcal{W}$ is the corresponding writer identity of x), i.e.,

$$\mathcal{L}_{id}^{D} = \mathbb{E}_{x,w}[-w \log I(x)], \quad (7)$$

when maximizing the adversarial loss. This ensures that I can distinguish which writer the image x belongs to. To further learn the disentangled style feature $\tilde{s} = E(x)$ from a reference image x, we enforce that the synthetic image $G(\tilde{\mathbf{y}}, E(x))$ to have a remarkably similar style compared with the reference image x. This is achieved by minimizing

$$\mathcal{L}_{id}^{G} = \mathbb{E}_{x,w,\tilde{\mathbf{y}}}[-w \log I(G(\tilde{\mathbf{y}}, E(x)))], \quad (8)$$

where I keeps fixing its parameters when minimizing the adversarial loss. This $\mathcal{L}_{id}^{G}$ essentially constrains the encoder E to produce close style embeddings for the images of similar styles, while diversifying the style embeddings for that of distinct styles (e.g., the handwriting images from different writers). Notably, the encoder E can disentangle the style form any reference image x without the need of accessing the corresponding writer identity w. Therefore, such an encoder E can potentially extract calligraphic styles from the handwriting images of unseen writers at test time.

Lastly, we further explicitly regularize the encoded latent space to match the prior random distribution as

$$\mathcal{L}_{kl} = \mathbb{E}_x[D_{KL}(E(x)||\mathcal{N}(0,1))], \quad (9)$$

where D_{KL} denotes the KL-divergence (Zhu et al. 2017).

Full Objective. Our full objective functions can be summarized as follows. When maximizing the adversarial loss, the discriminator D, recognizer R, and writer identifier I are optimized respectively as

$$\mathcal{L}_D = -\mathcal{L}_{adv},\ \mathcal{L}_R = \mathcal{L}_{ctc}^{D},\ \mathcal{L}_I = \mathcal{L}_{id}^{D}. \tag{10}$$

When minimizing the adversarial loss, the generator G and style encoder E are jointly optimized as

$$\mathcal{L}_{G,E} = \mathcal{L}_{adv} + \lambda_{ctc}\mathcal{L}_{ctc}^{G} + \lambda_{id}\mathcal{L}_{id}^{G} + \lambda_{sty}\mathcal{L}_{sty} + \lambda_{kl}\mathcal{L}_{kl}, \tag{11}$$

where λ controls the importance of different losses. Lastly, all modules are trained from scratch in an end-to-end way.

Experiments

Implementation Details. Inspired by Fogel (2020), all hyper-parameters λ are dynamically adjusted via the rule of gradient balance during training. The model is optimized using Adam (Diederik and Ba 2015) with a learning rate of 0.0001 and $(\beta1, \beta2) = (0.5, 0.999)$. Experiments are conducted on a Dell workstation with an Intel(R) Xeon(R) CPU E5-2630 v4@2.20GHz, 32 GB RAM, and NVIDIA Quadro P5000 GPU 16GB. The batch size is set to 16 for all experiments, and all models are trained over 100K iterations.

Baselines. We use GANwriting (Lei Kang 2020) and ScrabbleGAN (Sharon Fogel and Litman 2020) as our baselines, all of which can learn to generate handwritten words/texts. Briefly, GANwriting can synthesize short handwritten words conditioned on referenced styles, and ScrabbleGAN can synthesize long sentences with random styles. Differently, our HiGAN can synthesize handwriting images conditioned on arbitrary-length texts and disentangled styles. More details of the competing GANs can refer to Table 1. Lastly, all baselines are trained using the official implementations provided by the authors.

Datasets. To evaluate our HiGAN, we use the following two handwriting benchmarks:

- **IAM** (Marti and Bunke 2002) dataset consists of 9862 text lines with around 63K English words, written by 500 different writers. The dataset provides the official splits with mutually exclusive authors. In our settings, only the training & validate sets are used for training GANs.
- **CVL** (Florian Kleber and Sablatnig. 2013) dataset consists of seven handwritten documents (one German and six English texts) with about 83K words, written by 310 writers. The dataset is officially divided into the training set (with 27 writers) and test set (with 283 writers).

In our experiments, **IAM** is mainly used for training HiGAN, while **CVL** is only for handwriting recognition tasks.

Evaluation Metrics. For image synthesis, we use Fréchet Inception Distance (FID) (Martin Heusel and Hochreiter 2017) to evaluate the quality and diversity of synthetic images, where FID measures the distance between the generated distribution and real one through features extracted via Inception-v3 network (Christian Szegedy and Wojna 2016). For handwriting recognition, we use the character error rate (CER) and word error rate (WER) (Graves et al. 2008) to evaluate the handwriting recognition performance.

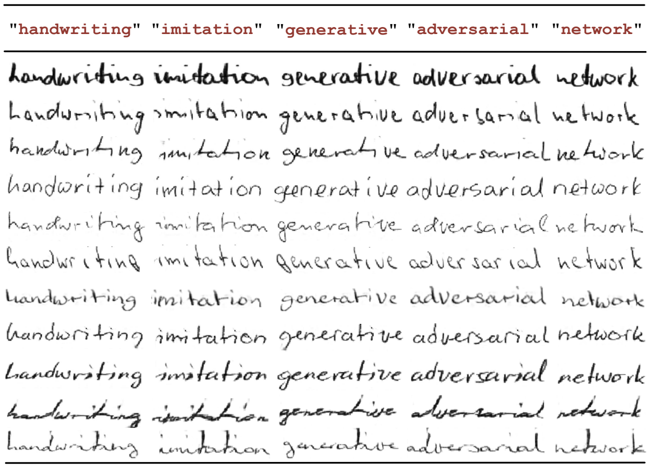

"handwriting" "imitation" "generative" "adversarial" "network"

Figure 3: Latent-guided synthesis. Random styles of synthetic images are sampled from a prior normal distribution.

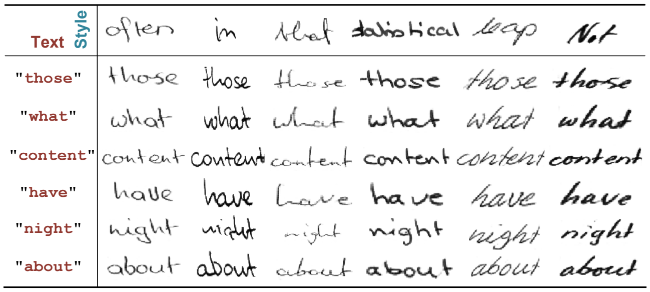


Figure 4: Reference-guided synthesis. Different styles of synthetic images are disentangled from reference samples.

Experimental Results

Latent-Guided Synthesis. Our HiGAN model is capable of generating arbitrary-length handwritten words with diverse calligraphic styles. For latent-guided synthesis, different styles of synthetic images are randomly sampled from a prior normal distribution (refer to Fig. 2(d)). Specifically, Fig. 3 shows some selected synthetic images of different words with various styles, where each row presents handwriting with the same style and each column with the same text. It is worth noting that HiGAN can render cursive ligatures between adjacent characters if necessary.

Reference-Guided Synthesis. HiGAN can disentangle calligraphic styles from reference images, and it further imitates generating images of similar styles with other textual contents. Fig. 4 illustrates some selected synthetic samples under reference-guided synthesis. It can be observed that HiGAN successfully imitates calligraphic styles that similar to the reference samples (e.g., the word slant & skew, stroke

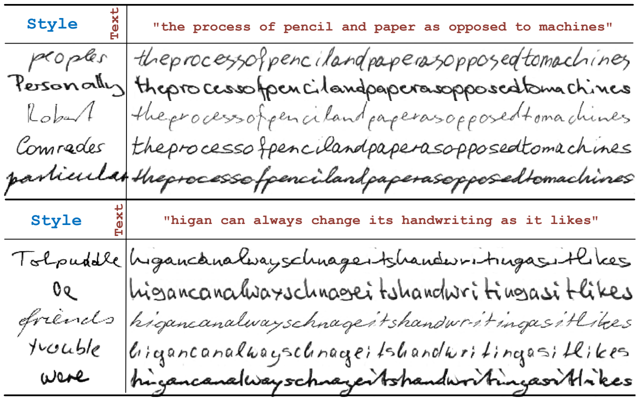

Style
Text
"the process of pencil and paper as opposed to machines"
Style
Text
"higan can always change its handwriting as it likes"

Figure 5: Long handwritten text generation. For the provided textual information, we omit all space letters and join all words together to form a long text string.

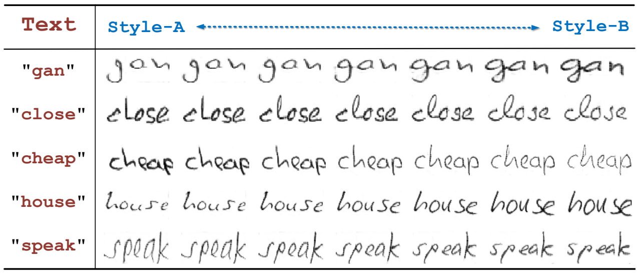


Figure 6: Style interpolation between two different styles.

width, and character shape), while mostly matches the provided textual contents. This result demonstrates that the proposed HiGAN is capable of imitating the calligraphic styles that disentangled from reference handwriting images.

Long Text Synthesis. Although our model is only trained with short words, HiGAN can still synthesize arbitrary-length long texts (as shown in Fig. 5). This merit is credited to the superior architecture design of generator. Specifically, a long text is first transformed into multiple character filter-maps and then concatenated into a variable-length text-map. With the fully convolutional structure, the style-invariant text-map then is upsampled into the target image conditioned on the given style, under which the character overlaps & cursive ligatures are automatically learned with convolutions. Notably, HiGAN even can synthesize long texts of similar styles that disentangled from only short words.

Style Interpolation. To further investigate the latent style space of HiGAN, we perform the linear interpolation between two random styles (as shown in Fig. 6). It can be observed that the handwriting image continuously changes its calligraphic style conditioned on the interpolation values, while strictly preserving its original textual contexts. This result demonstrates that our model can generalize in the style space rather than memorizing some trivial visual information.

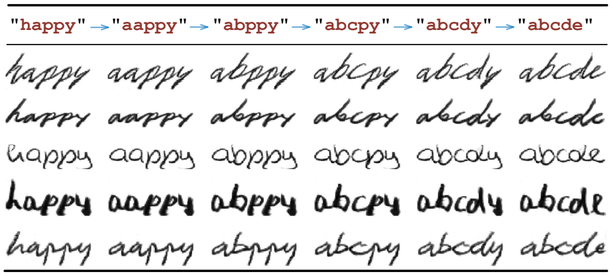


Figure 7: Handwritten text editing. From “happy” to “abcde”, the word changes one letter each time, while strictly preserving its original calligraphic style.

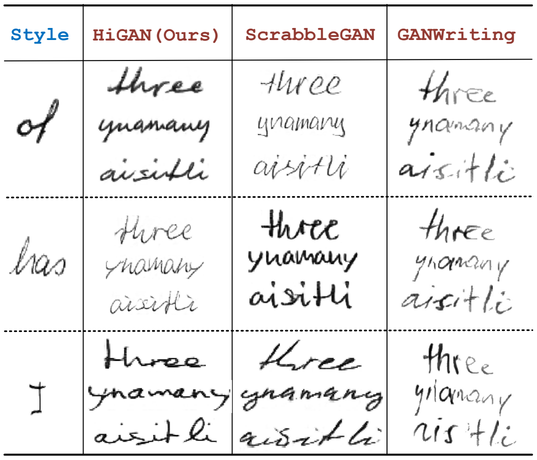


Figure 8: Qualitative comparison of different GANs for synthesizing short handwritten words. In addition, the words for each style are: “three”, “ynamany”, and “aisitli”, where the last two are not valid English words.

Text Editing. In contrast to style interpolation, we perform the text editing between two words (as shown in Fig. 6), where the interpolation is done in text space. As observed, when the word continuously changes its text letter-by-letter into another word, it strictly preserves its original calligraphic style. Moreover, our model not only renders natural ligatures among adjacent characters after replacing the specific letter, but it also successfully generates OOV words with high visual quality (e.g., “abcde” is not a valid English word). This result demonstrates that HiGAN can generalize in text space rather than memorizing training words/texts.

Compared with the State-of-Art GANs. To demonstrate the superiority of HiGAN, we first make a qualitative comparison of the competing GANs for synthesizing short handwritten words (in Fig. 8) and long texts (in Fig. 9), respectively. As shown in Fig. 8 & 9, each GAN produces multiple outputs with various styles but fixed textual contents. However, we notice that ScrabbleGAN cannot imitate the calligraphic styles of reference samples, since it lacks an encoder to disentangle styles from handwriting images. Moreover, GANwriting suffers from low visual quality when the

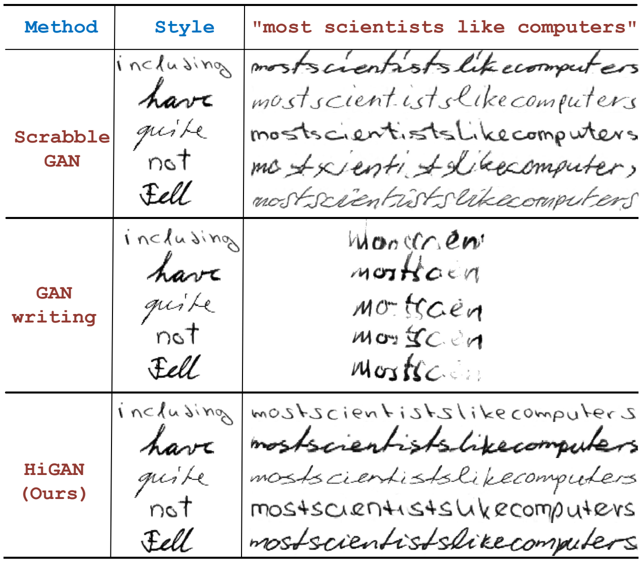


Figure 9: Qualitative comparison of different GANs for synthesizing long handwritten texts. For given texts, we omit all spaces and then concatenate all words into a long text string.

Method	Score	Size (MB)		
	FID	Gen.	Enc.	Disc.
GANwriting *Kang* (2020)	120.07	97.6	76.5	78.7
ScrabbleGAN *Fogel* (2020)	23.78	81.8	×	138.7
HiGAN-L.(*Ours*)	*19.80*	**38.6**	**20.5**	**35.2**
HiGAN-R.(*Ours*)	**17.28**			

Table 2: Quantitative comparison of GANs in terms of visual quality and model size. In addition, "HiGAN-L." and "HiGAN-R." denote the latent-guided and reference-guided syntheses of HiGAN respectively. Lower values are better.

style feature is encoded from only one reference image (as shown in Fig. 8). This is because GANwriting requires multiple reference images to extract a reliable style feature for each synthetic sample during training. Thus, the visual quality of synthetic image will degrade if only one style image is provided at test time. What is worse, GANwriting even cannot generate long handwritten texts (i.e., it can only generate words with less than ten letters) as shown in Fig. 9. On the contrary, HiGAN can not only generate diverse handwriting images of arbitrary-length texts but also imitate the calligraphic styles of reference images.

Lastly, we make a quantitative comparison of different GANs in Table 2. Note, FID scores of the competing GANs are borrowed from the original papers, and their evaluation strategies may be slightly different. We observe that HiGAN outperforms the competing GANs in terms of visual quality and model size. Overall, experiments demonstrate the superiority of HiGAN for handwritten word/text synthesis.

Training Data			CVL (%)		CVLoov (%)	
GAN	IAM	CVL	WER	CER	WER	CER
×	√	×	40.92	21.91	46.23	23.69
√	√	×	*40.73*	*20.87*	*45.36*	*22.10*
×	×	√	**32.69**	**14.71**	**41.72**	**20.25**
×	√	√	29.41	13.13	37.63	17.16
√	√	√	**28.91**	**12.54**	**37.06**	**16.67**

Table 3: Recognition results on CVL dataset. Notably, the first three rows show the results of domain adaption, and the last row shows that of semi-supervised training. In addition, the highlighted row is the upper bound of the domain adaption from IAM to CVL. It is also worth noting that HiGAN is only trained on the IAM dataset. Lower values are better.

Improving Recognition with GAN. We further demonstrate that HiGAN can improve the handwriting recognition performance of domain adaption in a semi-supervised manner. To validate this, we compare the recognition performance with or without HiGAN for training set augmentation (as shown in Table 3). In our experimental settings: (1) HiGAN is only trained on IAM and then utilized to synthesize fake handwriting images from an open lexicon; (2) for the CVL dataset, only English scripts are used for training; (3) despite English words from CVL, we also include the German words (only which share the English alphabet) as out-of-vocabulary words at test time, thus resulting in two CVL test sets (i.e., CVL and CVLoov); (4) during training, HiGAN is used for training set augmentation, and CRNN (Baoguang Shi and Yao 2016) is for handwriting recognition. As shown in Table 3, we observe that HiGAN can boost the recognition performance of domain adaption from IAM to CVL, and this improvement is more apparent for out-of-vocabulary words in CVLoov. For the semi-supervised training, we can further slightly improve the recognition accuracy on CVL by using HiGAN to synthesize extra words from an open lexicon. Overall, experimental results demonstrate that a strong generative model can potentially benefit the handwriting recognition.

Conclusion

In this paper, we have proposed a novel HiGAN for handwriting imitation. Our model can generate diverse and realistic handwriting images conditioned on arbitrary textual contents, which are unconstrained to any predefined corpus and OOV words. Moreover, HiGAN can disentangle styles from reference samples and flexibly control the handwriting styles of synthetic images. Furthermore, we also find that HiGAN can potentially benefit handwriting recognition by augmenting training set with extra synthetic data. Both qualitative and quantitative comparisons validate our superiority over the competing GANs in terms of visual quality and scalability. However, the human handwriting style is very arbitrary and thus HiGAN indeed exists a limit to synthesize meaningful handwriting images. In future work, we plan to further improve the diversity and visual quality of HiGAN.

Acknowledgments

This work is supported by National Key R&D Program of China under Grant No.2017YFB1002203; National Nature Science Foundation of China (NSFC) under Grant No.61976201; and NSFC Key Projects of International (Regional) Cooperation and Exchanges under Grant No.61860206004; and Ningbo 2025 Key Project of Science and Technology Innovation under Grant No.2018B10071.

Ethics Statement

We believe that our work will have broad positive implications in the development of artificial intelligence (AI). Specifically, our generative model can synthesize massive realistic labelled handwriting images after learning from a limited number of collected data. This can potentially benefit the construction of handwriting recognition systems that aim at serving our society. Moreover, our work can further teach computers/robotics to write scripts as realistic as humans, thus stepping closer to the high-level AI. On the other side, one potential negative ethical impact of our work is that it may be used for handwriting forgery. However, such a concern may be overestimated, because (1) our generative model essentially exists a limit to synthesize meaningful handwriting images since the human handwriting style is very arbitrary, and (2) it is still not strong enough to fool the handwriting identification experts.

References

Achint Oommen Thomas, A. R.; and Govindaraju, V. 2009. Synthetic Handwritten CAPTCHAs. *Pattern Recognition* 42(12): 3365–3373.

Alec Radford, Luke Metz, S. C. 2013. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *arXiv preprint arXiv:1511.06434*.

Alex Graves, S. F.; and Gomez, F. 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *International Conference on Machine Learning*, 369–376.

Andrew Brock, J. D.; and Simonyan, K. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference for Learning Representations*.

Baoguang Shi, X. B.; and Yao, C. 2016. An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(11): 2298–2304.

Bo Chang, Qiong Zhang, S. P.; and Meng, L. 2018. Generating Handwritten Chinese Characters using CycleGAN. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*.

Christian Szegedy, Vincent Vanhoucke, S. I. J. S.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Diederik, K. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference for Learning Representations*.

Eloi Alonso, B. M.; and Messina, R. 2019. Adversarial Generation of Handwritten Text Images Conditioned on Sequences. In *International Conference on Document Analysis and Recognition*, 481–486.

Florian Kleber, Stefan Fiel, M. D.; and Sablatnig., R. 2013. Cvl-Database: An Offline Database for Writer Retrieval, Writer Identification and Word Spotting. In *International Conference on Document Analysis and Recognition*, 560–564.

Graves, A. 2013. Generating Sequences with Recurrent Neural Networks. In *arXiv preprint arXiv:1308.0850*.

Graves, A.; Liwicki, M.; Fernández, S.; Bertolami, R.; Bunke, H.; and Schmidhuber, J. 2008. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(5): 855–868.

Harm de Vries, Florian Strub, J. M. H. L. O. P.; and Courville, A. C. 2017. Modulating Early Visual Processing by Language. In *Advances in Neural Information Processing Systems*, 6594–6604.

Hsin-Ying Lee, Hung-Yu Tseng, J.-B. H. M. S.; and Yang, M.-H. 2018. Diverse Image-to-Image Translation via Disentangled Representations. In *Proceedings of the European conference on computer vision*, 35–51.

Ian Goodfellow, Jean Pouget-Abadie, M. M.-B. X. D. W.-F. S. O. A. C. Y. B. 2014. Generative Adversarial Nets. In *Advances in neural information processing systems*, 2672–2680.

Jun-Yan Zhu, Taesung Park, P. I.-A. A. E. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision*, 2223–2232.

Kingma, D. P.; and Welling, M. 2013. Auto-Encoding Variational Bayes. In *arXiv preprint arXiv:1312.6114*.

Lei Kang, Pau Rib, Y. M. R.-A. F. M. V. 2020. GANwriting: Content-Conditioned Generation of Styled Handwritten Word Images. In *Proceedings of the European conference on computer vision*.

Lin, Z.; and Wan, L. 2007. Style-Preserving English Handwriting Synthesis. *Pattern Recognition* 40(7): 2097–2109.

Marti, Z.-V.; and Bunke, H. 2002. The IAM-Database: An English Sentence Database for Offline Handwriting Recognition. *International Journal on Document Analysis and Recognition* 5(1): 39–46.

Martin Heusel, Hubert Ramsauer, T. U. B. N.; and Hochreiter, S. 2017. GANs Trained by a Two Time-scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*, 6626–6637.

Mirza, M.; and Osindero, S. 2014. Conditional Generative Adversarial Nets. In *arXiv preprint arXiv:1411.1784*.

Sharon Fogel, Hadar Averbuch-Elor, S. C. S. M.; and Litman, R. 2020. ScrabbleGAN: Semi-Supervised Varying Length Handwritten Text Generation. In *Proceedings of the IEEEConference on Computer Vision and Pattern Recognition*, 4324–4333.

Tom S. F. Haines, O. M. A.; and Brostow, G. J. 2016. My Text in Your Handwriting. *ACM Transactions on Graphics* 35(3): 1–18.

Yunjey Choi, Youngjung Uh, J. Y. J.-W. H. 2020. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8188–8197.

Yunjey Choi, Minje Choi, M. K. J.-W. H. S. K.; and Choo, J. 2018. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8789–8797.

Zhu, J.-Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A. A.; Wang, O.; and Shechtman, E. 2017. Toward Multimodal Image-to-Image Translation. In *Advances in Neural Information Processing Systems*.