HiGAN+: Handwriting Imitation GAN with Disentangled Representations

JI GAN, Chongqing University of Posts and Telecommunications and University of Chinese Academy of Sciences
WEIQIANG WANG, University of Chinese Academy of Sciences
JIAXU LENG and XINBO GAO, Chongqing University of Posts and Telecommunications

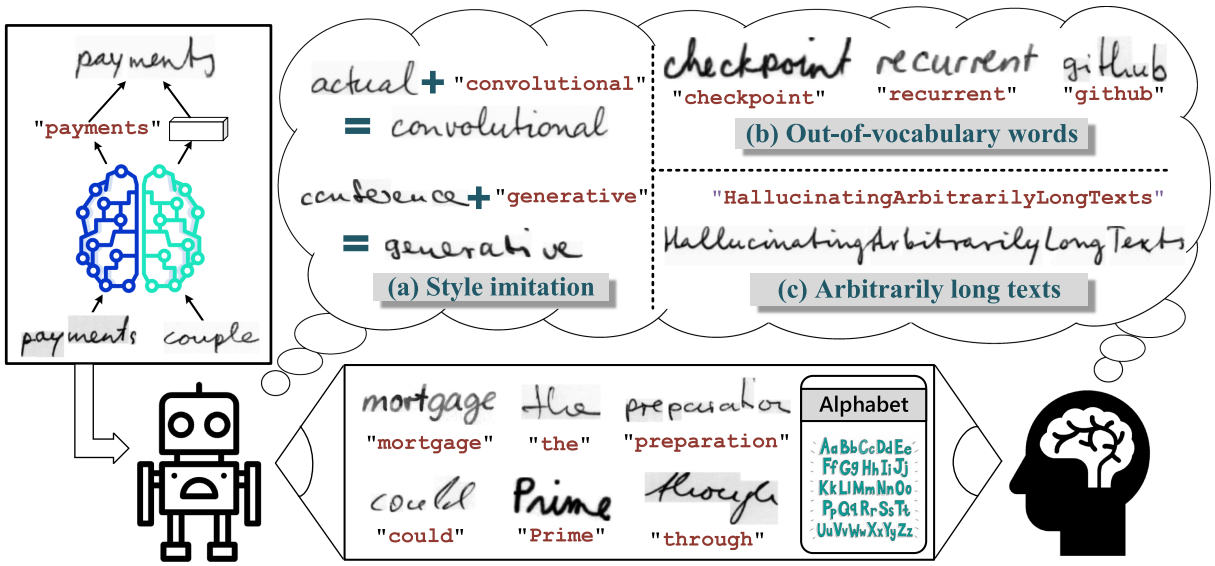


Fig. 1. Humans can quickly learn handwriting imitation with the ability of hallucination, while this task may be challenging for machines. Our goal is to teach machines to mimic such hallucinations, so that they may write diverse and realistic texts as well as humans after learning from limited handwriting scripts.

Humans remain far better than machines at learning, where humans require fewer examples to learn new concepts and can use those concepts in richer ways. Take handwriting as an example, after learning from very limited handwriting scripts, a person can easily imagine what the handwritten texts would like with other arbitrary textual contents (even for unseen words or texts). Moreover, humans can also hallucinate to imitate calligraphic styles from just a single reference handwriting sample (that even have never seen before). Humans can do such hallucinations, perhaps because they can learn to disentangle the textual contents and calligraphic styles from handwriting images. Inspired by this, we propose a novel handwriting imitation generative adversarial network (HiGAN+) for realistic handwritten text synthesis based on disentangled representations. The proposed HiGAN+ can achieve a precise one-shot handwriting style transfer by introducing the writer-specific auxiliary loss and contextual loss, and it also attains a good global & local consistency by refining local details of synthetic handwriting images. Extensive experiments, including human evaluations, on the benchmark dataset validate our superiority in terms of visual quality, scalability, compactness, and style transferability compared with the state-of-the-art GANs for handwritten text synthesis.

CCS Concepts: • **Computing methodologies** → **Image representations;**

Additional Key Words and Phrases: Handwriting imitation, handwritten text generation, generative adversarial networks, machine learning

ACM Reference format:
Ji Gan, Weiqiang Wang, Jiaxu Leng, and Xinbo Gao. 2022. HiGAN+: Handwriting Imitation GAN with Disentangled Representations. *ACM Trans. Graph.* 42, 1, Article 11 (September 2022), 17 pages.
https://doi.org/10.1145/3550070

This work is supported by the National Nature Science Foundation of China (NSFC) under Grant No. 61976201, No. 62036007, No. 62101084, and No. 62102057; in part by the NSFC Key Projects of International (Regional) Cooperation and Exchanges under Grant No. 61860206004, the Special Project on Technological Innovation and Application Development under Grant No. cstc2020jscx-dxwtB0032, and Chongqing Excellent Scientist Project under Grant No. cstc2021ycjh-bgzxm0339.
Authors' addresses: J. Gan, Chongqing University of Posts and Telecommunications, Chongqing 400065, China and University of Chinese Academy of Sciences; email: ganji15@mails.ucas.ac.cn; J. Leng and X. Gao (corresponding author), Chongqing University of Posts and Telecommunications, Chongqing 400065, China; emails: {lengjx, gaoxb}@cqupt.edu.cn; W. Wang (corresponding author), University of Chinese Academy of Sciences, Beijing 101408, China; email: wqwang@ucas.ac.cn.

0730-0301/2022/09-ART11 $15.00
https://doi.org/10.1145/3550070

1 INTRODUCTION

Although machines can easily recognize humans' handwriting scripts with recent advanced techniques, it still remains challenging for machines to synthesize realistic handwriting images. Hence, it will step closer to high-level artificial intelligence if we can teach machines/robotics to write texts as realistic as humans. Generally, **handwriting imitation** (**HI**) aims at (1) synthesizing diverse and realistic handwriting images conditioned on arbitrary textual contents and (2) imitating the calligraphic styles of reference handwriting images (e.g., the character shape, stroke thickness, writing slant, and ligature). As shown in Figure 1, humans can quickly learn such HI with the ability of hallucination. Specifically, after learning from very limited handwriting scripts, humans can easily visualize (or imagine) what the handwritten texts would look like with the other arbitrary textual contents. Moreover, after providing a reference handwriting sample, humans can also hallucinate novel handwriting images of similar calligraphic styles yet with different textual contents. Lastly, humans also are able to write arbitrarily long words and even complete sentences or paragraphs. By rethinking humans' learning ability, the reason why humans can do such advanced hallucinations is perhaps that they can learn to disentangle calligraphic styles and textual contents from handwriting images (rather than simply memorizing the training samples). Therefore, if we can teach machines to mimic this learning process, they may learn HI as well as humans.

It has witnessed many great achievements in the field of image generation with the recent advances in **generative adversarial networks** (**GANs**) [Goodfellow et al. 2014] and **variational auto-encoders** (**VAEs**) [Kingma and Welling 2013]. By integrating the advantages of GANs and VAEs, computers nowadays are capable of synthesizing diverse and realistic nature images or oil paintings, and they can even perform image-to-image translation by learning the mapping between different visual domains. However, a significant observation is that HI as a special image synthesis task has not been fully explored yet. Particularly, we demonstrate that HI is substantially different from the **conventional image generation** (**CIG**) studied in previous works, mainly due to the following aspects:

(1) **Variable-Sized Outputs**. CIG mainly focuses on producing fixed-sized images, while HI requires generating variable-sized images since handwritten texts can be arbitrarily long (e.g., handwritten sentences typically are longer than handwritten words). Therefore, the generator for HI should be specifically designed for variable-sized outputs.
(2) **Arbitrary Textual Contents**. CIG can only generate images conditioned on predefined classes and thus is impossible to produce images for other unseen classes. However, HI requires textual contents to be more precise (i.e., exact characters in desired orders), which is expected to generate arbitrary handwriting images conditioned on arbitrary textual contents that are unconstrained to any pre-defined corpus or **out-of-vocabulary** (**OOV**) words (i.e., the words that have never been seen during training).
(3) **Different Style Transfer**. CIG aims at synthesizing nature images or oil paintings, whose styles can be modeled as dense textures (which can be effectively captured by Gram matrices). In contrast, handwriting images contain little textures since they mainly consist of a sparse set of continuous graphical elements (i.e., handwriting strokes and cursive ligatures). Moreover, humans' handwriting can be very arbitrary, and thus handwriting images may not be perfectly spatially aligned even though their styles are visually similar. What is worse, HI has stronger semantic constraints, under which the generated images may contain completely different textual contents with different spatial sizes. Therefore, it poses challenges for traditional style transfer methods based on pixel correspondence (e.g., Pix2Pix [Isola et al. 2017] and CycleGAN [Kim et al. 2017]).

Table 1. Feature-by-Feature Comparison of GANs for Handwritten Text Generation

Method	Text Length	Style Transfer		Refine Detail	Size↓ (MB)	Quality FID↓
		Shot	Acc.↑			
ScrabbleGAN	Arbitrary	×	×	×	81.8	26.78
TS-GAN	Arbitrary	One	0.05	×	172.1	33.90
GANwriting	Short	Few	0.16	×	135.8	20.55
HWT	Arbitrary	Few	0.17	×	131.3	19.69
HiGAN	Arbitrary	One	0.33	×	59.1	18.30
HiGAN+	Arbitrary	One	**0.42**	√	**21.7**	**5.95**

In the table, ↑ indicates that higher values are better, and vice versa.
Bold indicates the best result.

Those characteristics make HI could be more challenging than CIG.

Recently, several efforts have been made for handwritten text synthesis based on GANs, while none of them have successfully solved the aforementioned difficulties at the same time. Specifically, Alonso et al. [2019] first proposed to adopt GANs for synthesizing handwritten text images conditioned on the whole embeddings of entire words. However, this method can only synthesize fixed-sized images and also produce low visual qualities for OOV words. Fogel et al. [2020] proposed an improved method called ScrabbleGAN, which can synthesize arbitrarily long handwritten texts by concatenating all the letter-tokens. However, the major limitation is that ScrabbleGAN cannot imitate the calligraphic styles of reference samples and thus fails to control the styles of synthetic images. Furthermore, Kang et al. [2020] proposed a few-shot style-conditioned handwritten word generation GAN, i.e., GANwriting. However, GANwriting is limited to synthesizing short words (e.g., less than 10 letters) rather than long texts due to its inferior architectural design. Moreover, Bhunia et al. [2021] proposed HWT to synthesize handwritten texts with Transformers. However, both GANwriting and HWT require multiple reference samples for extracting reliable calligraphic features during training, thus exhibiting low visual qualities when only one reference sample is available in inference. Recently, Davis et al. [2020] proposed **text and style conditioned GAN** (**TS-GAN**) for handwritten text synthesis. TS-GAN can learn to extract styles from images based on the pixel-to-pixel reconstruction loss, while it fails to correctly imitate styles of reference samples in most cases. This is because handwriting images are not spatially aligned and contain few textures, which makes pixel correspondence ineffective to model calligraphic styles. In summary, the state-of-the-art GANs have not entirely solved HI yet.

To address the above challenges, in our previous conference work [Gan and Wang 2021], we have proposed a novel **handwriting imitation GAN** (**HiGAN**) for HI, which can generate diverse and realistic handwriting images conditioned on arbitrary textual contents (that are unconstrained to any predefined corpus or OOV words) and calligraphic styles (that are disentangled from reference samples). However, HiGAN may produce blurred and distorted characters, exhibiting low visual qualities of synthetic images. The presented HiGAN+ not only significantly improves the visual qualities of synthetic images but also achieves a more accurate handwriting style transfer with desired properties. Table 1 shows a feature-by-feature comparison between different GANs for handwritten text generation.

Overall, the presented work supposes a significantly extended version of our previous conference paper. Specifically,

(1) We enhance our prior HiGAN with several new technical contributions, including:
   — The contextual loss is introduced to improve the style consistency and achieves a better calligraphic style transfer.
   — The local patch refinement is proposed to improve the local consistency of synthetic images with higher visual qualities.
   — We derive a more compact and effective architecture by reusing the writer identifier for style encoding.
(2) We propose comprehensive metrics to fully measure the performance of GANs for variable-length handwritten text synthesis. Particularly, the newly proposed **writer identification error rate** (**WIER**) can quantitatively measure the handwriting style transferability of GANs, which has never been investigated before.
(3) We conduct more extensive experiments (including Turning tests) on benchmarks to fully compare the proposed HiGAN+ with other state-of-the-art GANs for HI, where HiGAN+ achieves the best performance in terms of visual quality, scalability, compactness, and style transferability.

2 RELATED WORK

2.1 Handwriting Synthesis

Traditional approaches for handwritten text generation not only involve expensive manual intervention for clipping glyphs and tagging individual characters, but they also require a strong domain-specific knowledge for modeling glyph layouts and rendering ligatures and background textures. For example, Haines et al. [2016] proposed such an algorithm that can render desired English texts in a specific writer's handwriting. Similarly, Lin and Wan [2007] proposed to compute features from individual glyphs and words based on geometric statistics and further learn to synthesize complete words/sentences with hand-crafted hierarchical rules. Of course, such manual interventions are extremely expensive, and their generalization and scalability are also limited due to the domain-specific knowledge and hierarchical rules.

With the great successes of deep learning techniques in computer vision and machine learning, artificial neural networks have been gradually used for handwriting synthesis. Specifically, Graves [2013] first proposed to synthesize online handwriting trajectories of English texts based on **recurrent neural networks** (**RNNs**), which can predict the future stroke points with Gaussian mixture models. Moreover, Ha and Eck [2018] proposed SketchRNN for synthesizing hand-drawn sketches. Furthermore, Zhang et al. [2018] successfully adopted this architecture to draw realistic online handwritten Chinese characters of thousands of categories. More recently, Kotani et al. [2020] proposed the decoupled style descriptor model for handwriting, which factors both character- and writer-level styles and thus synthesizes more realistic handwriting trajectories. However, such an RNN-based model is hard to learn long-range dependencies of long sequences, and also their generation is time-consuming since RNNs remain less amenable to parallelization. More lethally, it is challenging to collect massive trajectories in a natural setting, since their recordings require unique equipment like stylus pens and touch screens. Instead, it is much easier to collect handwriting images with ubiquitous cameras and scanners in our real lives. Hence, it is more practical to synthesize handwriting images rather than trajectories.

GANs have achieved much progress in many image synthesis tasks, including handwritten character generation. Specifically, Goodfellow et al. [2014] proposed to generate realistic handwritten digits by introducing the adversarial loss, and Kingma and Welling [2013] proposed a VAE instead. Furthermore, Mirza and Osindero [2014] proposed **conditional GANs** (**cGANs**) to constrain the handwriting generation conditioned on desired class labels. Moreover, Chen et al. [2016] proposed InfoGANs to address the model collapse and Radford et al. [2013] proposed **deep convolutional GANs** (**DCGANs**) to improve the generation capability, thus producing more diverse and realistic images. Particularly, Chang et al. [2018] successfully adopted CycleGANs [Kim et al. 2017] for synthesizing handwritten Chinese characters of thousands of categories. Moreover, many researches also adopted GANs for the glyph font generation [Azadi et al. 2018; Jiang et al. 2019; Cha et al. 2020; Park et al. 2021], which aims at generating fixed-sized and isolated glyph font characters (instead of long text strings) with desired styles. Overall, existing works mainly focus on synthesizing fixed-sized handwritten digits/characters, while handwritten text synthesis is rarely explored.

As an emerging research topic, only a few efforts have been made for synthesizing handwritten text images. Specifically, Alonso et al. [2019] first proposed a GAN-based model to synthesize handwritten words conditioned on the whole embeddings of input texts, while their model is limited to generating fixed-sized images and also produces low visual qualities for OOV words. Moreover, Fogel et al. [2020] proposed ScrabbleGAN which can generate arbitrary-length handwritten texts by concatenating letter-tokens together but fails to imitate calligraphic styles of reference samples. Furthermore, Kang et al. [2020] proposed GANwriting that can generate handwritten words conditioned on extracted calligraphic features in a few-shot setting and textual contents of a pre-defined text length. In their follow-up work [Kang et al. 2021], they further demonstrated that the use of realistic synthetic texts at training is useful for improving the handwritten text recognition performance. Moreover, Bhunia et al. [2021] proposed to synthesize handwritten texts with Transformers. Recently, Davis et al. [2020] proposed a TS-GAN for handwritten text synthesis, which learns to extract styles based on pixel-to-pixel reconstruction loss. However, TS-GAN fails to correctly

imitate styles of reference samples most of the time. In our prior work [Gan and Wang 2021], we proposed HiGAN that can synthesize variable-sized handwriting images conditioned on arbitrary-length texts and disentangled styles, while it sometimes produces blurred textures and distorted characters. Nevertheless, the state-of-the-art GANs have not entirely solved the text- and style-conditioned handwritten text synthesis yet.

2.2 GAN-Based Style Transfer

Computers nowadays can solve many image translation tasks by combining the conceptions of GANs and VAEs, and those tasks aim at transferring the style characteristics of a style image to a content image. Specifically, Isola et al. [2017] proposed a cGAN (i.e., pix2pix) for image-to-image translation, and Zhu et al. [2017] proposed BicycleGAN to enable more diversified outputs. Nevertheless, those methods all require paired training data. To address this problem, many works relax the dependency on paired data by leveraging the cycle consistency, e.g., CycleGAN [Kim et al. 2017], DIRT [Lee et al. 2018], MUNIT [Huang et al. 2018], and StarGAN [Choi et al. 2018]. Moreover, Mao et al. [2019] proposed the mode seeking regularization to ensure the output diversity, and Iizuka et al. [2017] proposed to refine local details of images. Essentially, this kind of style transfer is achieved by minimizing differences between the generated and target images, where pixel-to-pixel reconstruction is utilized for spatial alignments and Gram matrices for texture statistics. Similar techniques have been extended to related applications such as font synthesis [Gao et al. 2019], scene texts [Wu et al. 2019], caricature [Cao et al. 2018], and face editing [Portenier et al. 2018].

2.3 Glyph Font Synthesis

Glyph font synthesis aims at designing and generating glyph font images automatically, and it has witnessed great achievements in recent years [Azadi et al. 2018; Gao et al. 2019; Jiang et al. 2019; Cha et al. 2020; Wang et al. 2020; Park et al. 2021]. However, we demonstrate that the glyph font synthesis is largely different from HI in the following aspects:

(1) **Annotation Difficulty:** The font style transfer requires laborious annotations for supervision, such as paired training samples (i.e., the input images and corresponding pixel-level aligned ground-truth images) or even attribute annotations for attribute editing [Wang et al. 2020]. Instead, HI only imposes writers' identities to specify the calligraphic styles, which avoids laborious annotations. This task learns the style transfer more implicitly than font generation.
(2) **Characters vs. Strings:** Previous font generation can only generate single isolated characters; however, HI aims at synthesizing long handwritten text strings with variable-sized outputs and arbitrary textual contents that are unconstrained to any predefined corpus and OOV words.
(3) **Style Variations:** Font generation aims at designing fonts for the industry, and the glyph fonts have very small intra-category variations (i.e., the font of the specific character class and style always has a standard template); instead, humans' handwriting is very arbitrary and their writing styles vary significantly (e.g., a person even is hard to write the exactly same sentences twice with pixel-to-pixel correspondence). Different from glyph fonts with limited styles, a thousand people have a thousand different handwriting styles.

Indeed, the research on glyph font generation may bring some inspiration for HI.

2.4 Differences between the Prior and Presented Works

HI is a new research topic, and the state-of-the-art GANs have not entirely solved this challenging problem yet. In our previous work [Gan and Wang 2021], we have proposed a novel HiGAN for HI, which can generate handwriting images conditioned on arbitrary-length texts and any calligraphic styles of reference samples. However, the prior work is very preliminary, still leaving a big room for improvement. Specifically,

(1) **Generation Quality:** The prior HiGAN sometimes may produce blurred and distorted characters, exhibiting low visual qualities of synthetic images. In HiGAN+, we introduce a **local patch loss** (**LPL**) to refine the local details of synthetic images, which significantly improves the local consistency of synthetic images. This strategy effectively prevents HiGAN+ from producing blurred patches or distorted characters, thus leading to much higher visual qualities of synthetic handwritten texts. Moreover, considering the characteristics of handwriting images, we introduce the contextual loss dedicated to HiGAN+ to effectively model calligraphic styles, which significantly improves the style consistency and achieves a better handwriting style transfer.
(2) **Model Compactness:** The prior HiGAN requires training two individual modules (i.e., the writer identifier and style encoder). Instead, HiGAN+ derives a more compact and more effective architecture by rethinking the roles of individual modules in the current framework, i.e., reusing the early layers of the writer identifier for style encoding. In contrast to existing works, this strategy can avoid using a huge pre-trained VGG backbone or training an additional style encoder.
(3) **Comprehensive Evaluation:** The evaluations of prior work are weak and insufficient in some aspects, since it only evaluates the **Fréchet Inception Distance** (**FID**) and **word error rate** (**WER**) scores of HiGAN, GANwriting and ScrabbleGAN. Instead, the presented work proposes comprehensive metrics for HI, including (a) **Inception Score** (**IS**), FID, **Kernel Inception Distance** (**KID**), **Peak Signal to Noise Ratio** (**PSNR**), and **Mean Structural Similarity** (**MSSIM**) for visual quality, (b) WER for readability, and (c) the newly proposed WIER for style transferability. Especially, none of the previous works have ever attempted to quantitatively evaluate the handwriting style transferability before. Furthermore, more extensive experiments are conducted on benchmark datasets to demonstrate the superiority of HiGAN+ over many other state-of-the-art GANs (including ScrabbleGAN, GANwriting, TS-GAN, HWT, and HiGAN). Moreover, the presented work even conducts Turing tests for HI.
(4) **State-of-The-Art Performance:** Experimental results show that the presented framework significantly outperforms the prior work, and the proposed HiGAN+ achieves the

state-of-the-art performance for HI in terms of visual quality, scalability, compactness, and style transferability compared with the state-of-the-art GANs for handwritten text generation.

Overall, the presented HiGAN+ not only significantly improves the visual qualities of prior HiGAN but also achieves the more accurate handwriting style transfer with desired properties, supposing a significantly extended version of our prior work.

3 METHODOLOGY

3.1 Problem Formulation

We aim at teaching machines to synthesize diverse and realistic handwriting images conditioned on arbitrary textual content $\mathbf{y} = [y_1, \ldots, y_L]$ (with a length of L) and any calligraphic style s, i.e., $\hat{x} = G(\mathbf{y}, s)$, where G is the generator. Notably, the textual content $\mathbf{y}$ for handwriting generation can be very arbitrary, which is unconstrained to any predefined corpus or OOV words. Moreover, the conditioned calligraphic style s can be either (1) randomly sampled from a prior normal distribution $\mathcal{N}(0, 1)$ or (2) disentangled from the reference image x, i.e., $s = E(x)$, where E denotes the style encoder. As a result, our generative model can not only generate arbitrary handwritten texts with randomized styles, but it also is able to imitate the calligraphic styles of reference samples. Figure 2 illustrates the overview of the proposed HiGAN+.

3.2 Network Architecture

3.2.1 Style-Controlled Handwritten Text Generator. Different from conventional image synthesis tasks, handwritten text synthesis needs to generate variable-length images (instead of fix-sized ones) conditioned on arbitrary textual contents (even for unseen texts and OOV words). By revisiting humans' handwriting process, one major observation is that handwriting essentially is a local process (which is firstly introduced by ScrabbleGAN [Fogel et al. 2020]). More specifically, humans typically finish a handwriting text by writing its letters sequentially and individually, under which the character shapes and cursive ligatures are mostly influenced by their neighbor characters in a local range. Inspired by this, our generator is designed to mimic such a writing process. Briefly, rather than generating handwriting based on a single embedding of the entire text, the generator converts the text into character embeddings individually and then concatenates those local character patches together into a complete handwritten text, where the convolutions are utilized to learn the overlaps and transitions among characters. Overall, the style-controlled handwritten text generator is designed with the following two strategies:

Textual Content Embedding. Instead of encoding the entire textual content $\mathbf{y}$ into a fixed-sized representation, we prefer to learn the character-level embeddings of $\mathbf{y}$ and concatenate letter-tokens into a complete text. The reason for doing this is to improve the generalization ability of the generative model, under which the generation can be conditioned on arbitrary texts that are unconstrained to the training corpus or any OOV words (e.g., words that have never been seen during training). Specifically, let $\mathcal{A}$ be the alphabet and $\mathcal{F} = \{f_c | c \in \mathcal{A}\}$ be the set of character filter maps (where f_c is the embedding of the character c). To achieve the character-level embedding, the given textual content $\mathbf{y} = [y_1, \ldots, y_L]$ (with a length L) will be individually embedded into multiple filter maps as $\mathcal{F}(\mathbf{y}) = [f_{y_1}, \ldots, f_{y_L}]$. Moreover, each filter map can be further modulated with a consistent randomized noise vector ϵ to introduce subtle distortions for characters, i.e, $\mathcal{F}(\mathbf{y}, \epsilon) = [f_{y_1} \otimes \epsilon, \ldots, f_{y_L} \otimes \epsilon]$. Lastly, those filter maps are concatenated horizontally into a variable-sized text map $\mathcal{M}$, which can be regarded as a style-invariant embedding of $\mathbf{y}$.

Calligraphic Style Rendering. Given the style-invariant text map $\mathcal{M}$, the generator G will up-sample its spatial resolution and simultaneously render the calligraphic styles. Particularly, the **conditional batch normalization** (**CBN**) [Vries et al. 2017] is utilized to inject the style feature s into the generator, thus explicitly affecting the calligraphic styles of synthetic images (such as the text slant, character shape, and stroke thickness). Moreover, the generator follows a fully convolutional structure to ensure the variable-length outputs. Due to the merits of convolutions, the generator can automatically learn the overlaps between adjacent characters and create smooth transitions (i.e., natural ligatures) if necessary. This eventually leads to the generator being able to synthesize arbitrarily long handwritten texts conditioned on arbitrary textual contents with controllable calligraphic styles.

3.2.2 Other Components. To achieve precise HI, we further introduce the following key components to assist in the training process of HiGAN+:

Global Discriminator. The global discriminator D learns a binary classification to determine whether an input image x is the real image from the training data or the fake image produced by the generator G. By grading the whole image, the discriminator D can verify the fidelity of synthetic images from a global perspective.

Patch Discriminator. The patch discriminator P can justify whether a given patch ψ^x is the one cropped from real images or fake images. Instead of grading the whole image, it will help refine the local texture details of synthetic images by verifying the patch fidelity.

Style Encoder. The style encoder E is supposed to disentangle the calligraphic styles from arbitrary handwriting images but without explicitly accessing extra clues including the writer identities and text labels. Additionally, the encoder E can map arbitrary-length handwriting images into fixed-sized latent vectors (i.e., the calligraphic style features) for HI.

Writer Identifier. The writer identifier I can distinguish which writer the input handwriting image x belongs to, and it aims to guide the generator to synthesize handwriting images conditioned on specific calligraphic styles. Notably, the identifier I can only identify handwriting images of seen writers in training data, while it cannot classify that of unseen writers at test time.

Text Recognizer. The text recognizer R should correctly predict the text label $\mathbf{y}$ of any handwriting image x. Particularly, although the recognizer R is only trained on real, labeled, handwriting images, it is supposed to guide the generator G to produce arbitrary readable handwriting images conditioned on arbitrary textual contents.

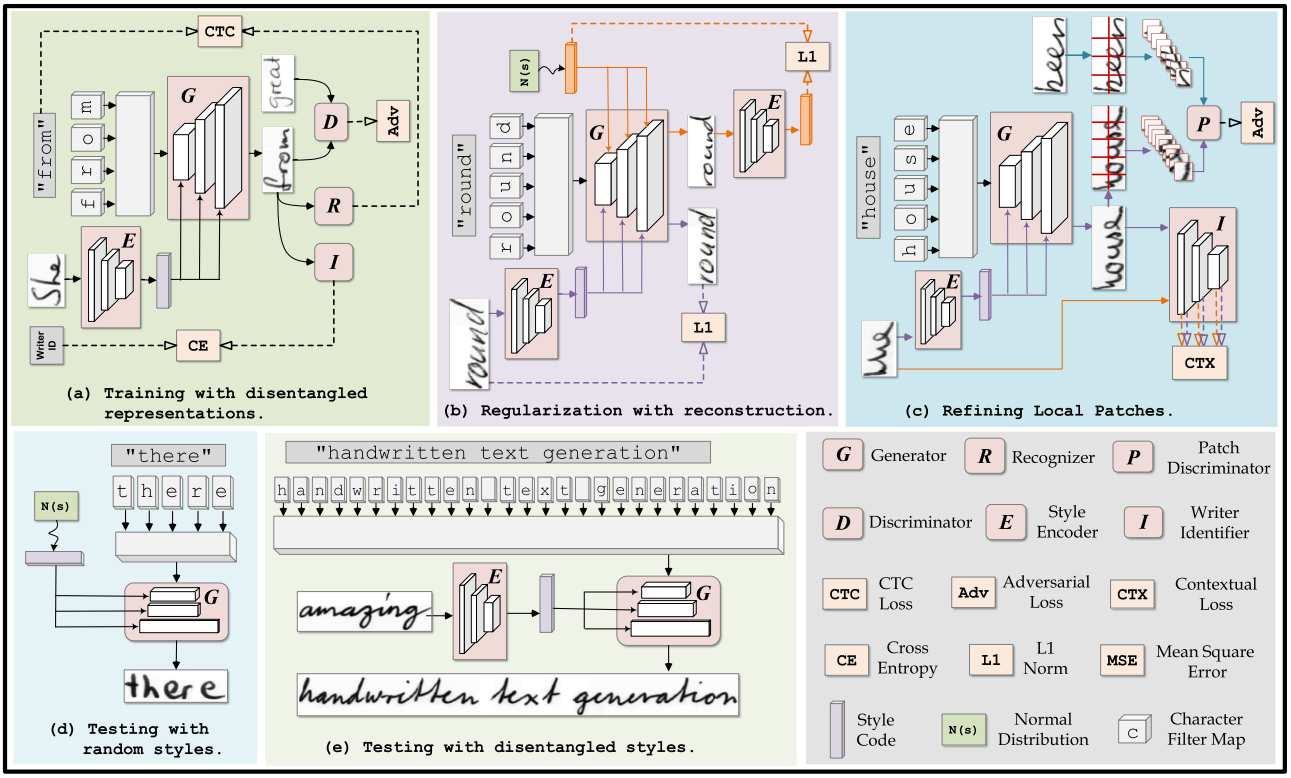


Fig. 2. Overview of the proposed HiGAN+. During training, the model simultaneously (a) learns to synthesize handwritten texts based on disentangled styles, (b) is regularized based on reconstruction, and (c) lastly refines the local details of synthetic images for improving visual qualities. At the test time, the model can either (d) generate diverse handwritten texts by randomly sampling styles from a prior normal distribution or (e) imitate the calligraphic styles that disentangled from reference samples. Notably, each module shares its parameters at different training stages.

3.3 Objective Functions

To train HiGAN+ for HI, it requires a multi-writer handwriting dataset which consists of the sets of handwriting images $\mathcal{X}$, their labeled texts $\mathcal{Y}$, and the corresponding writer identities $\mathcal{W}$. Since the handwritten text generation is not limited to the training corpus or OOV words, a large open corpus $\mathcal{C}$ is utilized to yield arbitrary textual contents during training, where $\mathcal{Y} \subset \mathcal{C}$. As shown in Figure 2, we illustrate the overview of the training process, and the details of different losses are formulated below.

3.3.1 *HI with Disentangled Representations.*

Adversarial Loss. Following the paradigm of GANs, the generative model is trained via a min-max adversarial game. During training, the generator G takes arbitrary textual content $\tilde{\mathbf{y}} \in \mathcal{C}$ and a style feature s as inputs and then learns to synthesize a fake image $G(\tilde{\mathbf{y}}, s)$ that is indistinguishable (by the discriminator D) from the real one $\mathbf{x} \in \mathcal{X}$ via the adversarial loss, i.e.,

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

where the style feature s is either (1) randomly sampled from a prior normal distribution $\mathcal{N}(0, 1)$ or (2) disentangled from the reference image x, i.e., $s = E(x)$. Notably, the adversarial loss only promotes the general visual appearance of generated images to make them look realistic, while it does not consider preserving either textual contents or calligraphic styles.

Text Recognition Loss. Despite visual appearances, the generator G is supposed to synthesize realistic readable handwriting images with preserving desired textual contents. To this end, a handwriting recognizer R is introduced to guide G toward producing handwriting images with specific textual contents. Specifically, the recognizer R is first optimized by theoretically maximizing the likelihoods for each pair $\{x \in \mathcal{X}, y \in \mathcal{Y}\}$ from the training data (where the connectionist temporal classification loss [Graves et al. 2006] is empirically adopted in HiGAN+) as

$$\mathcal{L}^D_{ctc} = \mathbb{E}_{x,\mathbf{y}}[-\mathbf{y} \log R(x)], \tag{2}$$

when maximizing the adversarial loss. This ensures that R can correctly predict the text labels of given handwriting images. Although the recognizer R is only trained with real, labeled, handwriting images, it is supposed to guide the generator G to synthesize readable handwriting conditioning to arbitrary textual content $\tilde{\mathbf{y}} \in \mathcal{C}$ as

$$\mathcal{L}^G_{ctc} = \mathbb{E}_{\tilde{\mathbf{y}},s}[-\tilde{\mathbf{y}} \log R(G(\tilde{\mathbf{y}}, s))], \tag{3}$$

where the parameters of R keep fixed when minimizing the adversarial loss.

Writer Identification Loss. The primary objective of HiGAN+ is to exactly disentangle the calligraphic styles from reference handwriting images and further imitate generating different images of similar styles but with other textual contents. However, one major

concern is that we actually do not have the exact labels for writing styles (including the stroke thickness, character shape, and text slant) to form the style consistency regularization. To avoid expensive manual annotations, we impose the writer identity to specify the calligraphic style. The reason for doing this is based on a simple assumption that the writing style of each individual is unique and almost consistent. Therefore, the writer identifier I is introduced to guide the encoder E in disentangling calligraphic styles from reference samples.

Specifically, the identifier I is optimized by minimizing the cross-entropy loss for each pair $\{x \in \mathcal{X}, w \in \mathcal{W}\}$ from training data as

$$\mathcal{L}_{id}^{D} = \mathbb{E}_{x,w}[-w \log I(x)], \tag{4}$$

when maximizing the adversarial loss. This ensures that I can identify which writer the reference image x belongs to. To guide the encoder E to exactly disentangle calligraphic styles from reference samples, we enforce the style-conditioned synthetic images $G(\tilde{\mathbf{y}}, E(x))$ to retain a remarkably similar style with the reference image x, i.e.,

$$\mathcal{L}_{id}^{G} = \mathbb{E}_{x,w,\tilde{\mathbf{y}}}[-w \log I(G(\tilde{\mathbf{y}}, E(x)))], \tag{5}$$

where I keeps its parameters fixed when minimizing the adversarial loss, and the textual content $\tilde{\mathbf{y}} \in C$ is not limited to the training corpus. It is worth noting that the identifier I is only trained on the training set and thus it is unable to identify the writers that have never been seen during training (i.e., the writers in the test set).

3.3.2 Regularization with Reconstruction.

Style Reconstruction Loss. To encourage an invertible mapping between synthetic images and style features, we apply a style reconstruction loss similar to Chen et al. [2016] as

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

where the style feature s is sampled from the prior normal distribution $\mathcal{N}(0, 1)$. This regularization loss essentially exhibits two advantages: (1) It guarantees that the style feature s can explicitly affect calligraphic styles of synthetic handwriting images; (2) It encourages the diversified outputs and thus helps avoid model collapses of the generative network.

Content Reconstruction Loss. To improve the content and style consistency of synthetic images, we adopt a self-reconstruction loss to facilitate the training, i.e.,

$$\mathcal{L}_{recn} = \mathbb{E}_{\mathbf{y},x}[||x - G(\mathbf{y}, E(x))||_1], \tag{7}$$

where $\mathbf{y} \in \mathcal{Y}$ is the labeled text of image x. Following this auto-encoding training scheme, it may regularize the generative model to achieve a more robust handwriting style transfer.

KL-Divergence Loss. To ensure a meaningful stochastic style sampling in inference, we further explicitly regularize the encoded latent space to match the prior normal distribution as

$$\mathcal{L}_{kl} = \mathbb{E}_{x}[D_{KL}(E(x)||\mathcal{N}(0, 1))], \tag{8}$$

where D_{KL} denotes the KL-divergence [Zhu et al. 2017]. This is a crucial regularization technique in many style transfer tasks [Zhu et al. 2017; Lee et al. 2018].

3.3.3 Local Detail Refinement.

Contextual Loss. Conventional style transfer is achieved by synthesizing an image to match both the contents and styles of target images, which commonly compares images in two aspects: (1) the pixel-to-pixel loss that compares pixel values at the same spatial coordinates; (2) the Gram loss that compares high layer features and texture information over the entire image. This method is very effective for nature images or oil paintings, since their styles are modeled as texture features. In contrast, handwriting images contain little textures and their styles are modeled as the character shape, thickness, and slant. Moreover, humans' handwriting can be very arbitrary and the synthetic handwriting may not be exactly spatially aligned with the ground-truth images, and thus handwriting images with similar styles may produce a large reconstruction loss. What is worse, the synthetic handwriting and reference samples may have completely different textual contexts and spatial sizes. Therefore, the conventional style transfer strategy is unsuitable for HI.

To address this problem, we introduce the contextual loss [Mechrez et al. 2018] to measure the similarity of two handwriting images, requiring no spatial alignment. The key idea of contextual loss is to treat an image as a collection of features, and the similarity between images is measured based on the similarity between their high-level features, ignoring the spatial positions of the features. This loss focuses more on high-level style features and allows the generated images to be slightly spatially deformed with respect to ground-truth images. Moreover, the contextual loss is not overly global and it compares features in local regions based on semantics. Let $\mathbf{A} = \{a_1, \ldots, a_N\}$ and $\mathbf{B} = \{b_1, \ldots, b_N\}$ be two sets of features, the contextual similarity between them is defined as

$$\mathrm{CX}(\mathbf{A}, \mathbf{B}) = \frac{1}{N} \sum_{j} \max_{i} \mathrm{CX}_{ij}, \tag{9}$$

where CX_{ij} denotes the similarity between features a_i and b_j, and CX_{ij} is calculated by normalizing all the cosine distances d_{ij} between any a_i and b_j as Mechrez et al. [2018]. In our task, we apply the contextual loss to achieve better HI, i.e.,

$$\mathcal{L}_{ctx} = \sum_{l} -log\mathrm{CX}\left(\Phi^{l}(x), \Phi^{l}(G(\tilde{\mathbf{y}}, E(x)))\right), \tag{10}$$

where $\Phi^{l}(\cdot)$ denotes the high-level features extracted from the lth layer of the writer identifier I, and $\mathrm{CX}(\cdot, \cdot)$ denotes the aforementioned contextual similarity between two feature sets.

Local Patch Loss. Handwritten text images can be arbitrarily long, and thus they can be regarded as high-resolution images. Although it can achieve a good global consistency (i.e., a synthetic image is globally visually plausible) by grading the whole image from a global perspective, such a strategy may lead to poor local consistency (i.e., the synthetic handwriting image may contain many blurred patches and distorted characters). Therefore, it is crucial to refine the local texture details of synthetic handwriting images. Despite classifying the whole image as fake or real, we further split each image into patches and then justify the patch fidelity by introducing an extra patch discriminator. The introduced patch discriminator can penalize the local structures and thus help achieve

good local consistency. Let $\{\psi_i^x | i = 1 \cdots M\}$ and $\{\psi_i^{\tilde{\mathbf{y}},s} | i = 1 \cdots M\}$ be the patches of real image x and generated one $G(\tilde{\mathbf{y}}, s)$ respectively, the local details of synthetic images are refined as

$$\mathcal{L}_{patch} = \frac{1}{M}\sum_{i=1}^{M}\left\{\mathbb{E}_x[\log P(\psi_i^x)] + \mathbb{E}_{\tilde{\mathbf{y}},s}[\log(1 - P(\psi_i^{\tilde{\mathbf{y}},s}))]\right\}, \tag{11}$$

where P is the patch discriminator. It is worth noting that our patch discriminator receives image patches as inputs rather than the entire image, and thus it is not limited to the simple and specific network design of PatchGAN [Isola et al. 2017]. As a result, our patch discriminator can be more flexible and complex than that of PatchGAN, which eventually may lead to better synthesis performance.

3.3.4 Overall Objectives. Finally, our model is trained by playing a min-max adversarial game, where the full objective functions can be summarized as follows.

When maximizing the adversarial loss, the global discriminator D, patch discriminator P, text recognizer R, and writer identifier I are individually optimized as

$$\mathcal{L}_D = -\mathcal{L}_{adv}, \mathcal{L}_P = -\mathcal{L}_{patch}, \ \mathcal{L}_R = \mathcal{L}_{ctc}^D, \ \mathcal{L}_I = \mathcal{L}_{id}^D. \tag{12}$$

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

$$\mathcal{L}_{G,E} = \mathcal{L}_{adv} + \mathcal{L}_{patch} \tag{13}$$

$$+\lambda_{ctc}\mathcal{L}_{ctc}^G + \lambda_{id}\mathcal{L}_{id}^G + \lambda_{ctx}\mathcal{L}_{ctx} \tag{14}$$

$$+\lambda_{style}\mathcal{L}_{style} + \lambda_{recn}\mathcal{L}_{recn} + \lambda_{kl}\mathcal{L}_{kl}, \tag{15}$$

where λs are the hyper-parameters to control the importance of different loss terms.

3.4 Training Strategies

3.4.1 Pre-Training the Writer Identifier and Text Recognizer. For the writer identifier $\mathcal{I}$ and text recognizer $\mathcal{R}$, their optimization actually can be separated from the adversarial training process. Specifically, we can benefit from such a pre-training in two aspects:

(1) We can obtain more powerful and robust $\mathcal{I}$ and $\mathcal{R}$ by introducing data augmentation and extra handwriting samples during pre-training, since their optimization is separated from the adversarial training process.
(2) Once the $\mathcal{I}$ and $\mathcal{R}$ are pre-trained, the adversarial training of HiGAN+ can be further accelerated. Moreover, we can avoid retraining new $\mathcal{I}$ and $\mathcal{R}$ when training a different HiGAN+.

Finally, pre-training both $\mathcal{I}$ and $\mathcal{R}$ will not hurt the performance of HiGAN+ empirically.

3.4.2 Reusing Writer Identifier as Style Encoder. By rethinking the role of each component of HiGAN+ for HI, we propose to reuse the writer identifier as the style encoder as shown in Figure 3. Specifically, if we check the relations between the style encoder and writer identifier, their functions are almost consistent. Basically, to correctly identify handwriting images, the writer identifier should extract their calligraphic styles but ignore the semantic textual contents; Similarly, the style encoder is exactly designed to disentangle styles from handwriting images. Upon this motivation, it is intuitive to reuse the writer identifier for encoding

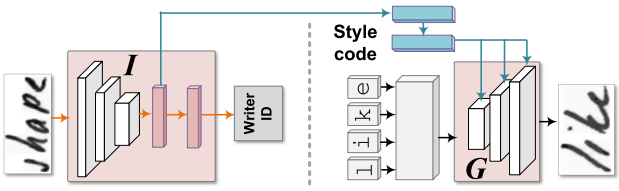


Fig. 3. Reusing writer identifier as style encoder.

styles. However, most previous works for handwriting synthesis simply use the huge VGG backbone as the style encoder, and they also train two separated modules (i.e., the writer identifier and style encoder) rather than merging them as ours. Those eventually make their model more parameter-redundant and computation-expensive than HiGAN+. In summary, such reusing exhibits two-fold advantages:

(1) We achieve a more compact architecture, since the style encoder and writer identifier share a large number of parameters. Moreover, this strategy can benefit the training procedure of HiGAN+, since lots of parameters of the style encoder are well pre-trained and thus it can extract more reliable style features at the early stage.
(2) We avoid using a huge VGG backbone for style encoding, which is pre-trained on nature images instead of handwriting images. In contrast, our writer identifier is specifically designed for handwriting identification, which is more suitable for extracting calligraphic styles and also attains a more compact model than the VGG backbone.

During training, the shared parameters of the style encoder keep fixed and only the independent part is optimized.

3.4.3 Optimization with Gradient Balancing. The loss function of HiGAN+ for optimization involves quite a few terms, and the relative weighting of different loss terms will affect the synthesis results. Moreover, it also takes lots of time to fully optimize the HiGAN+ for handwriting synthesis. Therefore, it may be difficult to find an optimal setting of those hyper-parameters λs for HiGAN+ with the naive grid search. To address this issue, we adopt the gradient balancing strategy to dynamically adjust the hyper-parameters λs of HiGAN+, thus balancing the gradient of each loss term to stabilize the training procedure and reach a satisfied local optimum.

Specifically, the gradient of $\mathcal{L}_{adv}$ warped on the synthetic image $\hat{x}$ is first calculated as

$$\nabla_{adv} = \frac{\partial \mathcal{L}_{adv}}{\partial \hat{x}}. \tag{16}$$

Take $\mathcal{L}_{ctc}$ for example, its gradient can be balanced [Alonso et al. 2019] by

$$\nabla_{ctc} \leftarrow \frac{\delta(\nabla_{adv})}{\delta(\nabla_{ctc})}[\nabla_{ctc} - \mu(\nabla_{ctc})] + \mu(\nabla_{adv}), \tag{17}$$

where $\mu(\nabla_{ctc})$ denotes the mean of ∇_{ctc} and $\delta(\nabla_{adv})$ denotes the standard deviation. To avoid changing the sign of the gradient ∇_{ctc}, we adopt a simpler strategy similar to Fogel et al. [2020], i.e.,

$$\nabla_{ctc} \leftarrow \frac{\delta(\nabla_{adv})}{\delta(\nabla_{ctc})}\nabla_{ctc}. \tag{18}$$

Therefore, the weight of $\mathcal{L}_{ctc}$ is adjusted as $\lambda_{ctc} = \frac{\delta(\nabla_{adv})}{\delta(\nabla_{ctc})}$ and other hyper-parameters can be calculated in the same way.

3.5 Evaluation of GANs for HI

HI is different from CIG due to its variable-sized outputs, arbitrary textual contents, and different style transfer. Hence, it is essential to quantitatively measure the qualities of synthetic handwriting images from different aspects. Therefore, we propose comprehensive metrics to fully evaluate the performance of GANs for variable-length handwritten text synthesis. Specifically,

- **Visual Quality:** Synthetic handwriting images should first deceive the human eyes visually and be realistic as far as possible. Therefore, we adopt several commonly used metrics to evaluate the visual quality of handwriting images, including *IS* [Salimans et al. 2016], *FID* [Heusel et al. 2017], *KID* [Binkowski et al. 2018], *PSNR*, and *MSSIM*. Particularly, IS is used to measure the realism and diversity of generated images, FID and KID aim to measure the distance between distributions of the generated images and real samples, MSSIM measures the structural similarity between them, and PSNR measures the reconstruction error.
- **Readability:** Different from natural images, handwriting images convey specific semantic information that can be read and understood by humans. Therefore, we use the *WER* to evaluate the readability of synthetic texts, which is the number of word recognition errors divided by that of total words. Particularly, the word recognition can be done by humans or a pre-trained handwriting recognizer.
- **Style Transferability:** Besides the realism and readability, the calligraphic styles of synthetic images should be consistent with the reference samples as much as possible. Therefore, we propose to use the *WIER* to measure the style transferability of GANs for HI, which is the number of writer identification errors divided by that of the total words. It is worth noting that none of the previous works have ever quantitatively evaluated the style transferability of GANs for HI.

Since the handwritten texts are variable-length instead of fixed-sized, we replace the averaging pooling of IncepetionV3 with **Temporal Pyramid Pooling** (**TPP**) when calculating IS, FID, and KID (similar to Kang et al. [2021]), and we also use the global averaging pooling in the CNN backbone of the writer identifier when calculating WIER. In our settings, all GANs are trained using the training set images and all evaluations are conducted on test set images. However, since the writers in test set have never been seen in training set, we need to train an additional writer identifier using the test images to evaluate the WIER. Such a writer identifier is entirely independent of the presented framework and thus can fairly evaluate the style transferability of different GANs for HI.

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 Datasets. To evaluate the performance of handwriting generation, we use the **IAM** dataset [Marti and Bunke 2002] as the benchmark dataset. The IAM dataset consists of 63K handwritten English words, written by 500 different writers. It provides one training set, one test set, and two validation sets. It is worth noting that handwritten words of all sets are mutually exclusive, thus each writer only contributes to one set. In our experiments, only the training set and validation sets are used for training GANs, and the test set is only used for quality evaluation.

4.1.2 Implementation Details. Our experiments are conducted on a Dell workstation with an Intel(R) Xeon(R) Bronze 3204 CPU @ 1.90 GHz, 48 GB RAM, and GeForce RTX 3090 GPU 24 GB. For fast training, the batch size is set to 8 and the model is trained for 70 epochs. Furthermore, we utilize the Adam [Diederik and Ba 2015] algorithm to optimize the GAN model, where the initial learning rate is 0.0001 and $(\beta_1, \beta_2) = (0.5, 0.999)$. Moreover, we begin to linearly decay the learning rate at the 25th epoch. When training HiGAN+, we empirically set $\lambda_{kl} = 0.0001$, $\lambda_{ctx} = 5.0$, and the rest λs are dynamically adjusted during training with the gradient balancing strategy. The training time is less than three days on a single GeForce RTX 3090 with our implementation in PyTorch [Paszke et al. 2019].

4.1.3 Competitors. Previous works mainly focus on handwritten character/digit generation, while handwritten text generation has not been fully explored. In our experiments, we can only compare our method with several recently proposed handwritten text generation approaches, i.e., **ScrabbleGAN** [Fogel et al. 2020], **GANwriting** [Kang et al. 2020], **TS-GAN** [Davis et al. 2020], **HTW** [Bhunia et al. 2021], and **HiGAN** [Gan and Wang 2021] (where Table 1 gives a detailed feature-by-feature comparison). We use the official implementation of those models provided by the authors, where we directly use the default settings and pre-trained models if available. Notably, we also retrain HiGAN with our new network configurations, which can generate handwriting images with a fixed height of 64 pixels rather than 32 pixels. In our experiments, all synthetic handwriting images are resized to have 64 pixel height while preserving the original aspect ratios. For a fair comparison, all evaluations are conducted on test set images. More specifically, we optimize all GANs with training set images and then use those generative models to reconstruct test set images.

4.2 Qualitative Analysis

In this subsection, we first conduct the qualitative analysis of HiGAN+ for arbitrary handwritten text generation.

4.2.1 Latent-Guided Synthesis. The proposed HiGAN+ can generate arbitrary handwritten English words of diverse calligraphic styles with high visual quality. For latent-guided synthesis, different styles of synthetic images are simply randomly sampled from the prior normal distribution. Specifically, we show some selected synthetic images in Figure 4, where each row presents images of the same styles and each column of the same texts. It is worth noting that all generated handwritten words are human-readable, and they are unconstrained to the predefined corpus or OOV words. Moreover, we observe that HiGAN+ can successfully render cursive ligatures among adjacent characters of handwritten words if necessary.

4.2.2 Reference-Guided Synthesis. For reference-guided synthesis, our HiGAN+ can precisely disentangle calligraphic styles

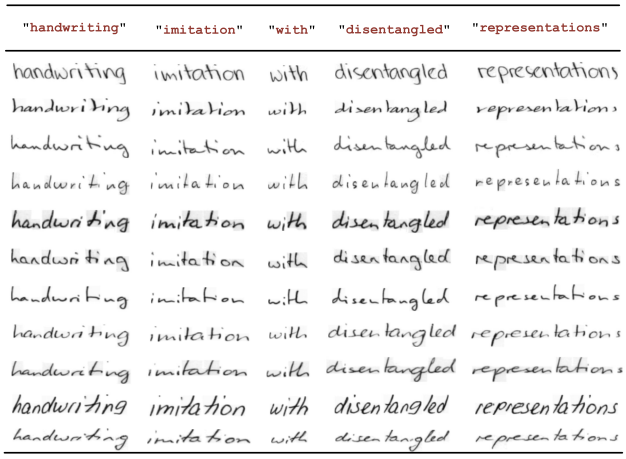


Fig. 4. Latent-guided synthesis.

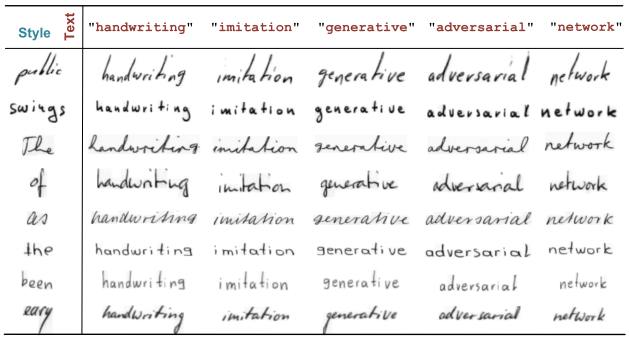


Fig. 5. Reference-guided synthesis.

from reference samples, and it further imitates generating other handwriting images of similar calligraphic styles. As shown in Figure 5, we show some selected handwritten words under reference-guided synthesis. We can observe that HiGAN+ can successfully imitate the calligraphic styles of reference samples (such as the writing slant, thickness, and character shape), while strictly preserving the desired textual contents. Overall, Our HiGAN+ can achieve precise one-shot handwriting style transfer.

4.2.3 Arbitrary-Length Text Synthesis. The proposed HiGAN+ can generate variable-sized images conditioned on arbitrary-length texts, which are unconstrained to any predefined corpus or OOV words. As shown in Figure 6, we show some selected long synthetic handwritten texts. Particularly, since handwritten English sentence generation can be easily accomplished by word generation, we omit all spaces of the provided textual content to form an extremely long text string for handwriting synthesis in Figure 6. Rather than generating handwriting images conditioned on a single embedding of the entire word/text, HiGAN+ will convert the text into character embeddings individually and then concatenate them together. Moreover, the generator with a fully convolutional network structure will automatically learn overlaps and ligatures among adjacent characters. Notably, HiGAN+ can even

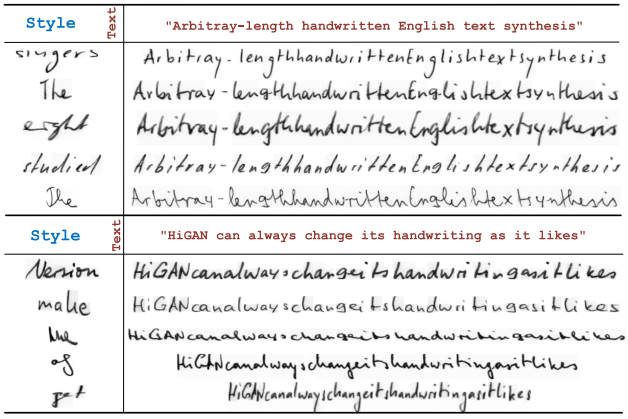


Fig. 6. Arbitrary-length text synthesis. Notably, all spaces of the provided textual content are omitted to form a long text string.

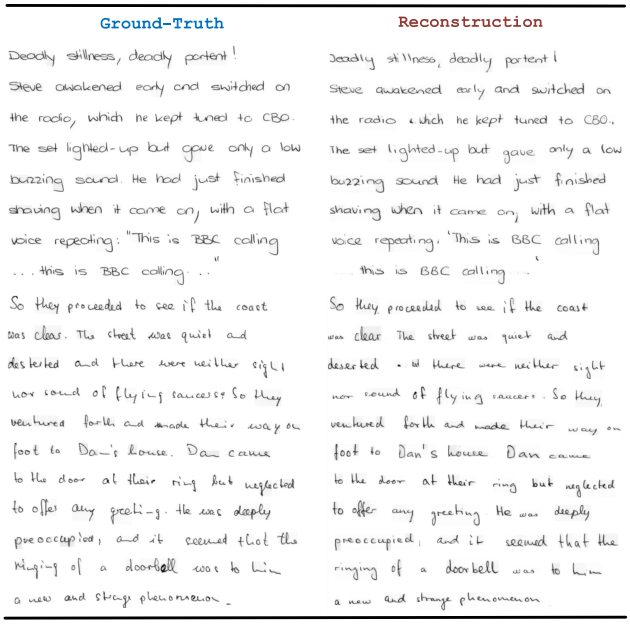


Fig. 7. Handwritten paragraph synthesis.

disentangle styles from short words to generate arbitrary-length text images of similar styles.

4.2.4 Handwritten Paragraph Synthesis. Despite words and texts, HiGAN+ can even generate complete handwritten paragraphs with its ability of one-shot style transfer for handwriting images. Figure 7 illustrates the original handwritten paragraph and the one reconstructed with HiGAN+, where each reconstructed word is synthesized based on the disentangled representations of the corresponding real word. We observe that HiGAN+ can successfully imitate calligraphic styles and preserve the original textual contents of most handwritten words. This eventually results in that the generated handwritten paragraphs look extremely realistic and mostly indistinguishable from the real ones. Overall, our

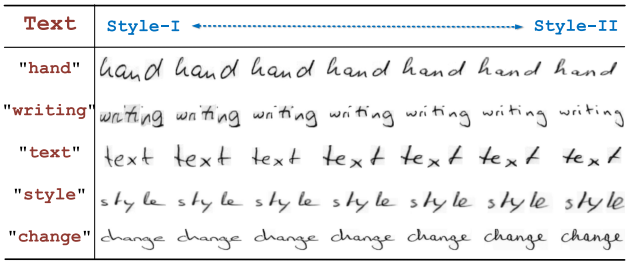


Fig. 8. Handwriting style interpolation.

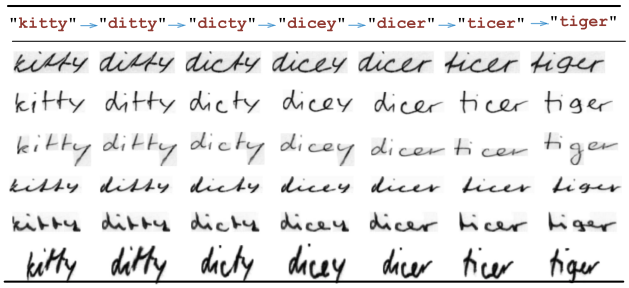


Fig. 9. Handwriting text editing.

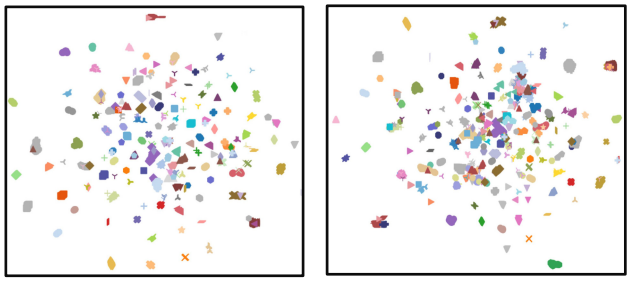
results demonstrate that HiGAN+ can perform precise one-shot HI with high visual qualities.

4.3 Generalization Analysis

We further investigate whether the generative model can imitate handwriting as humans, rather than simply memorizing the ground-truth images.

4.3.1 Style Interpolation. To better analyze the learned latent style space, we perform the linear interpolation between two random calligraphic styles and generate the corresponding handwriting images as shown in Figure 8. We can observe that the synthetic handwriting images continuously change their calligraphic styles (such as the thickness, character shape, and writing slant), while strictly preserving the original textual contents. Those results validate the continuity of the latent style space, thus demonstrating that HiGAN+ generalizes in the distribution rather than simply memorizing trivial visual appearances of training data.

4.3.2 Text Editing. Despite the latent style space, we also perform the interpolation in the text space to further validate the generalization of HiGAN+. In contrast to the continuous nature of the style distribution, the textual content space essentially is discrete. Therefore, we simply perform the handwriting text editing by following a "word ladder" puzzle game as shown in Figure 9, where we change the source word into the target one by replacing only one character at a time. We can observe that the synthetic handwriting images continuously change their textual contents, while strictly preserving the original calligraphic styles. Moreover, HiGAN+ not only draws the natural ligatures when replacing the specific letter but also successfully generates the OOV words (e.g., "kitty", "dicer", and "dicey"). The interpolation results validate that HiGAN+ can generate novel handwriting images that



(a) Ground Truth (b) Reconstruction

Fig. 10. The UMAP visualization of the latent vectors extracted by the encoder, where the shape and color identify the author.

unconstrained to any OOV words (rather than simply copying the training samples).

4.3.3 Style Embeddings. To further verify the generalization ability of HiGAN+, we show the UMAP visualization of the latent vectors extracted from both (a) test images and (b) reconstructed images of HiGAN+. As shown in Figure 10(a), the latent distributions (i.e., style features) of images from the same writer are clustered, while that of different writers are separated from each other. This demonstrates that HiGAN+ can cluster embeddings for handwriting images of similar styles and diversify embeddings for that of different styles, thus disentangling meaningful styles from handwriting images. Furthermore, we can also observe a similar phenomenon on the reconstruction images in Figure 10(b), which indicates that HiGAN+ can achieve a precise HI.

4.4 Ablation Studies

In this subsection, we conduct an ablation study to justify the contribution of each key component in HiGAN+. As shown in Table 2, we give the quantitative comparison of different configurations for handwritten text synthesis, where each component is cumulatively added on top of the baseline model.

Specifically, the baseline configuration (A) (with only the adversarial loss and CTC loss) corresponds to the basic setup of ScrabbleGAN [Fogel et al. 2020], which can only generate readable handwritten texts with randomized styles. As shown in the first row of Figures 11 and 12, the baseline model (A) fails to imitate the calligraphic styles of reference images, since it lacks a style encoder to disentangle handwriting styles. Moreover, the visual quality of synthetic images is poor, where some characters are even distorted and blurred as shown in Figure 11.

We first improve the baseline model (A) by regularizing the generator with reconstructing original image contents, i.e., the configuration (B) which is equivalent to the TS-GAN [Davis et al. 2020]. Additionally, we introduce the KL-Divergence loss to regularize the encoded latent space to match the prior normal distribution. With the explicit spatial alignment between synthetic images and ground-truth images, we wish the style encoder can extract meaningful style embeddings from reference images and thus is able to generate novel handwritten texts of similar styles. However, the WIER of configuration (B) is very high as shown in Table 2, which indicates that such an auto-encoding scheme cannot help the generative model to achieve precise HI. As shown in the second row

Table 2. Quantitative Comparison of Different Configurations for Handwritten Text Synthesis

ID	Method	IS↑	FID↓	KID↓	PSNR↑	MSSIM↑	WIER↓	WER↓
A	$\mathcal{L}_{adv}$ + $\mathcal{L}_{ctc}$	1.2975	24.3985	2.2173	11.6510	0.2078	1.0000	0.0243
B	A + $\mathcal{L}_{recn}$ + $\mathcal{L}_{kl}$	1.3356	25.2118	2.0649	11.3832	0.2452	0.9751	**0.0080**
C	B + $\mathcal{L}_{id}$ + $\mathcal{L}_{style}$	1.3298	18.3095	1.6688	11.7609	0.2459	0.6747	0.0085
D	C + $\mathcal{L}_{ctx}$	1.3694	10.6346	0.7752	11.9286	0.2981	0.6116	0.0085
E	D + D_{patch}	**1.4059**	**5.9510**	**0.3709**	**12.3391**	**0.3322**	**0.5821**	0.0186
	Ground Truth	1.4390					0.1105	0.1820

"↑" denotes that higher values are better, and "↓" denotes that lower values are better.
Bold indicates the best result.

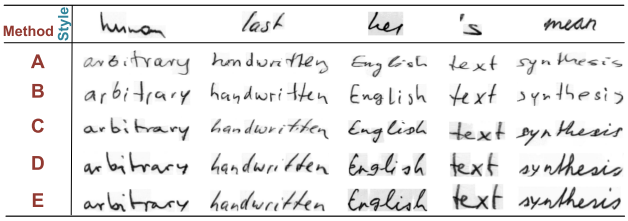


Fig. 11. Ablation study for handwritten words.

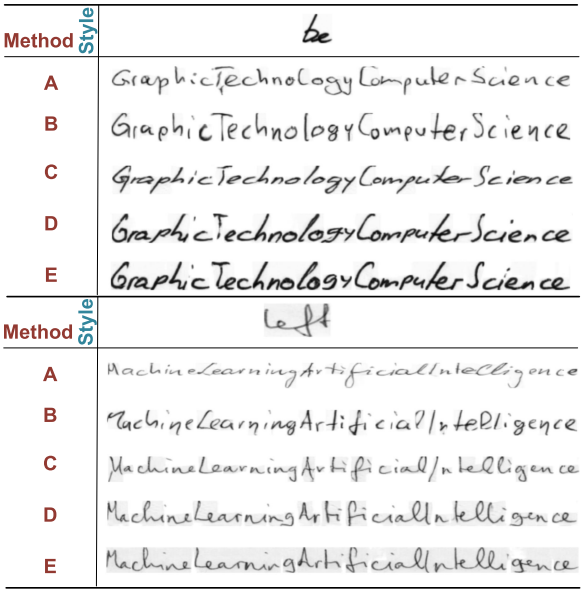


Fig. 12. Ablation study for long handwritten texts.

of Figures 11 and 12, the model (B) fails to imitate the calligraphic styles of reference images.

To achieve precise handwriting style transfer, we further introduce the writer identification loss $\mathcal{L}_{id}$ and style reconstruction loss $\mathcal{L}_{style}$, i.e., the configuration (C) which corresponds to HiGAN [Gan and Wang 2021]. Specifically, the term $\mathcal{L}_{id}$ can guarantee that the input style code can explicitly affect the styles of generated images. Furthermore, with the help of the writer identifier, the term $\mathcal{L}_{id}$ enforces the generator to synthesize images conditioned on a particular writer identity, e.g., that of reference images. Therefore, we can explicitly guide the generator to mimic the calligraphic styles of reference images. As shown in Table 2, the model (C) achieves much lower WIER and higher FID scores, which indicates that the model achieves more accurate calligraphic style transfer and significantly improves the visual quality of synthetic images.

We further improve the style consistency of synthetic images by introducing the contextual loss $\mathcal{L}_{ctx}$ on top of the model (C), i.e., the configuration (D). Since humans' handwriting is very arbitrary, it may be challenging to spatially align the synthetic handwriting images and ground-truth ones. Furthermore, in contrast to nature images, handwriting images contain little textures. Therefore, it may be insufficient to achieve precise handwriting style transfer with the conventional pixel-to-pixel reconstruction and Gram loss (that is designed for capturing textual features). However, the contextual loss can measure the style similarity between two images based on high-level feature map collections, requiring no spatial alignments. As shown in Table 2, the values of evaluation metrics clearly demonstrate the effectiveness of the contextual loss.

Lastly, we introduce the local patch loss $\mathcal{L}_{patch}$ on top of the model (D) to further refine the local texture details of synthetic images, i.e., the configuration (E) which corresponds to the proposed HiGAN+. Notably, handwritten text images can be arbitrarily long, and thus it cannot guarantee the local details for such high-resolution images. Instead of grading the whole image, we split each image into patches and then introduce another discriminator to justify the patch fidelity. As shown in Figures 11 and 12, the generative model without the LPL will produce blurred characters, and the $\mathcal{L}_{patch}$ term ensures that the generated images preserve better style consistency (e.g., grey textures in backgrounds). Finally, the results in Table 2 demonstrate that HiGAN+ significantly improves the visual quality and achieves a more precise calligraphic style transfer.

4.5 Comparison between PatchGAN and LPL

Although PatchGAN accepts the whole image and computes patches in parallel, its discriminator is limited to the specific shallow architectures to simulate the patch processing. For example, to simulate a patch processing with a path size of 32×32 and patch stride of 8, the deepest discriminator is limited to "$Conv2D(k = 3, s = 2) \rightarrow Conv2D(k = 3, s = 2) \rightarrow Conv2D(k = 3, s = 2) \rightarrow Conv2D(k = 3, s = 1)$" (where "$k$" is the kernel size and "s" is the stride); however, a deeper CNN will lead to a larger receptive filed (>32). In contrast, LPL physically splits the whole image into separated patches, and thus its patch discriminator is not limited to the specific architectural design and can be arbitrarily complex. Therefore, LPL with a more complex and powerful discriminator may achieve better performance than PatchGAN. To validate our assumption, we conduct a quantitative comparison between

Table 3. Comparison between PatchGAN and LPL

Method	Patch Discriminator			IS↑	FID↓	KID↓
	Type	Depth	Size			
Scrabble GAN	None	×	0.00	1.2975	24.3985	2.2173
	PGAN*	4	2.32	1.3367	15.2872	1.1609
	LPL*	8	2.26	**1.3607**	**13.0480**	**0.9890**
HiGAN	None	×	0.00	1.3298	18.3095	1.6688
	PGAN*	4	2.32	1.3521	13.0448	1.0290
	LPL*	8	2.26	**1.4393**	**12.0309**	**0.9795**

*"PGAN" is the PatchGAN and "LPL" is the LPL.

PatchGAN and LPL on the IAM dataset in Table 3, where their patch discriminators have the same number of parameters with the patch size of 32 × 32 and patch stride of 8. Experimental results show that LPL with more powerful patch discriminators may outperform PatchGAN.

4.6 Imitating Handwriting in the Wild

We show that HiGAN+ can also imitate calligraphic styles of handwriting images in the wild. Different from handwriting on the whiteboard, handwriting in the wild has more extreme and diverse calligraphic styles (including large variations in stroke thickness/colors and many noises and distortion of characters). Specifically, we have conducted experiments on a dataset for English handwriting in the wild named GNHK [Lee et al. 2021]. In our experiments, only the training set of GNHK is used for optimizing GANs and no other extra images are involved, and the test set is only used for evaluation. The qualitative results in Figure 13 show that HiGAN+ can synthesize handwriting images with more extreme handwriting styles. Lastly, we also give the quantitative results in Table 4, which demonstrates HiGAN+ significantly outperforms the baselines.

4.7 Comparison with the State-of-the-Arts

We compare the proposed HiGAN+ with recent state-of-the-art GANs for handwritten text synthesis. For all competing GANs, we use the official implementation with default settings and the pretrained models provided by the authors. For a fair comparison, we utilize all GANs to reconstruct the test set images of the IAM dataset.

4.7.1 Visual Comparison. As shown in Figures 14 and 15, we make a qualitative comparison between different GANs for handwriting synthesis to intuitively reflect their synthetic visual qualities. Notably, the original implementations of ScrabbleGAN, HTW, and HiGAN can only produce images with 32-pixel height, while GANwriting, TS-GAN, and HiGAN+ can produce images with 64-pixel height.

Although ScrabbleGAN can generate readable handwritten text images, it fails to imitate the calligraphic styles of reference samples. This is because ScrabbleGAN lacks a style encoder to disentangle calligraphic styles from images. Moreover, the visual qualities of its synthetic images are poor as many characters in handwritten texts are distorted and blurred.

Both GANwriting and HWT can control the calligraphic styles of synthetic handwriting images. For GANwriting, since it only

Fig. 13. Handwritten text synthesis in wild.

Table 4. Quantitative Results of Handwritten Text Synthesis in Wild on GNHK

Method	IS↑	FID↓	MSSIM↑	WIER↓	WER↓
ScrabbleGAN	1.4993	44.3732	0.2931	1.0000	0.2524
HiGAN	1.5291	26.8190	0.2940	0.7111	0.1751
HiGAN+	**1.6255**	**9.6546**	**0.3695**	**0.4480**	**0.1237**
Ground Truth	1.7592			0.0175	0.3602

encodes limited-length words to fixed-sized vectors, it cannot generate arbitrarily long handwritten texts (i.e., no more than 10 letters). As shown in Figure 15, GANwriting fails to complete the provided textual contents. For HTW, it utilizes the vision Transformer to capture the global and local styles of handwriting images. However, both HWT and GANwriting require multiple reference samples to extract reliable style features for HI.

For TS-GAN, it follows an auto-encoder architecture, which implicitly learns HI by reconstructing original images. Although TS-GAN successfully mimics thicknesses and text slants, it fails to mimic character shapes and texture backgrounds. Therefore, its ability for handwriting style transfer is limited, which demonstrates that the pixel-to-pixel reconstruction is insufficient for handwriting style transfer.

For HiGAN, it not only generates realistic handwritten texts but also successfully imitates calligraphic styles of reference samples. This is because HiGAN further introduces a writer-specific auxiliary loss to constrain the handwriting generation conditioned on particular writer identities. However, HiGAN sometimes produces a few distorted and blurred characters, since it only grades the whole image during training but fails to consider the local texture details.

For HiGAN+, we first introduce the contextual loss to improve the style consistency of HiGAN, which enhances the style similarity of images based on high-level feature map collections extracted by the writer identifier. Furthermore, we also refine the local

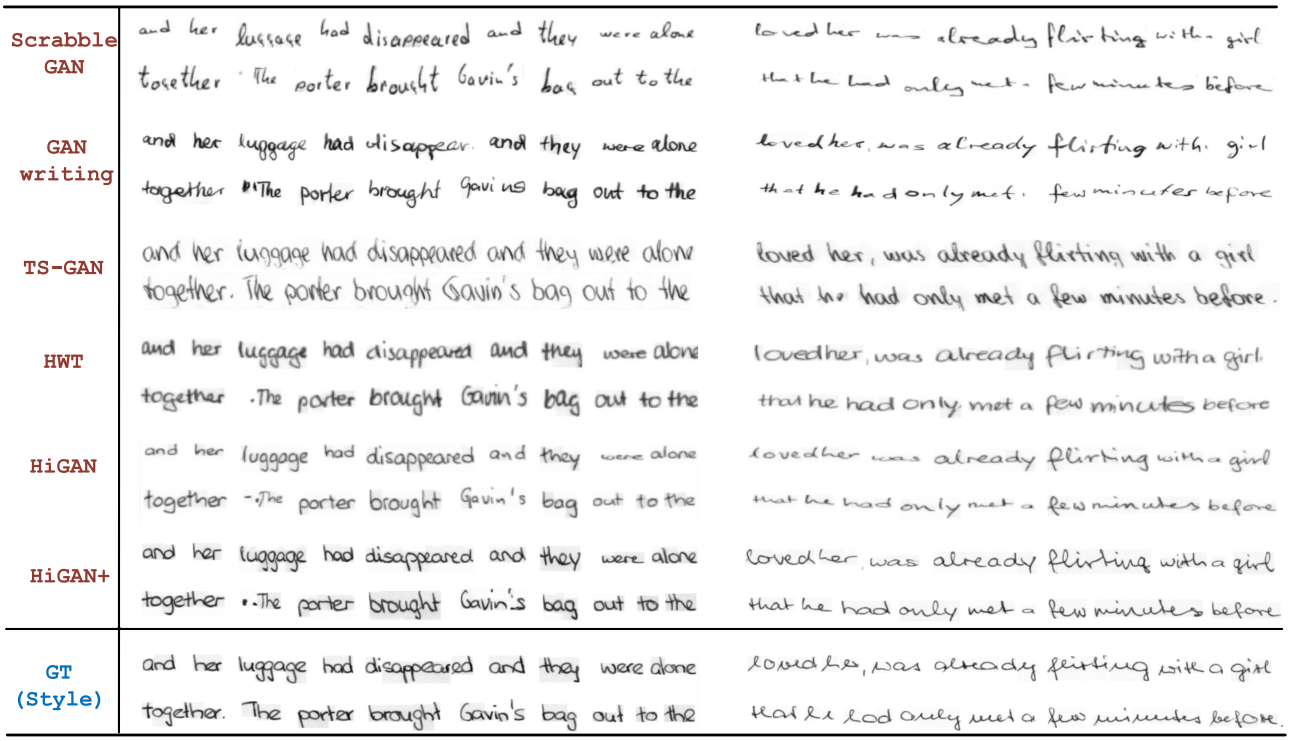


Fig. 14. Qualitative results of different GANs for handwritten paragraphs.

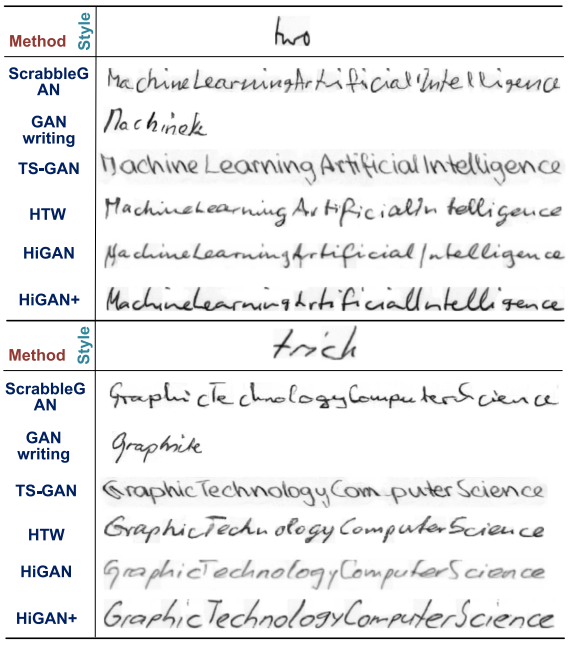


Fig. 15. Qualitative results of different GANs for long handwritten texts.

texture details of synthetic images by introducing a patch discriminator to verify the patch fidelity. Those strategies eventually make HiGAN+ attain a good global & local consistency. As shown in Figures 14 and 15, HiGAN+ produces clearer handwriting images and achieves a more precise HI.

4.7.2 Quantitative Evaluation. To give a higher-level indication of visual quality on the whole test set, we further conduct a quantitative evaluation between different GANs for handwriting synthesis as shown in Table 5. Moreover, we also have evaluated the metrics between the generated and real samples in different settings (in Table 6) including (1) in vocabulary and seen style (I-S), (2) in vocabulary and unseen style (I-U), (3) OOV and seen style (O-S), and (4) OOV and unseen style (O-U). We can observe that both ScrabbleGAN and ST-GAN obtain high FID and WIER, which indicates that they suffer from the poor visual quality and also fail to imitate the calligraphic styles of reference samples. Moreover, although HiGAN slightly outperforms GANwriting in terms of visual quality, it achieves more precise HI (i.e., lower WIER) and its synthetic images are much more readable (i.e., lower WER). Furthermore, the quantitative results in Table 5 clearly demonstrate that HiGAN+ largely outperforms the other state-of-the-art GANs for HI in terms of visual quality and it also achieves a more precise one-shot handwriting style transfer. Lastly, we list the model storages of different GANs in Table 7, and we can observe that the proposed HiGAN+ attains the most compact model for handwritten text synthesis compared with other state-of-the-art GANs. This is because HiGAN+ employs a compact style encoder (that is specifically designed for extracting handwriting styles) rather than using a huge pre-trained VGG backbone.

4.8 Failure Case Analysis

To investigate the weakness and limitation of the proposed HiGAN+, we conduct the failure case analysis as shown in Figure 16.

Table 5. Quantitative Comparison of Different GANs for Handwritten Text Synthesis

Method	IS↑	FID↓	KID↓	PSNR↑	MSSIM↑	WIER↓	WER↓
ScrabbleGAN	1.3268	26.7758	2.9479	11.2562	0.1950	1.0000	0.0740
ST-GAN	1.2443	33.9069	3.1314	12.0345	0.1845	0.9741	0.1968
GANwriting	1.3267	20.5539	1.3927	10.8045	0.2038	0.8455	0.2143
HWT	1.3620	19.6938	1.8003	10.7518	0.2319	0.8320	0.1032
HiGAN	1.3298	18.3095	1.6688	11.7609	0.2459	0.6747	**0.0085**
HiGAN+	**1.4059**	**5.9510**	**0.3709**	**12.3391**	**0.3322**	**0.5821**	0.0186
Ground Truth	1.4390					0.1105	0.1820

Table 6. Comparison of Different GANs in Different Settings for Handwritten Text Synthesis

	IS↑				FID↓				WIER↓				WER↓			
Method	I-S	I-U	O-S	O-U	I-S	I-U	O-S	O-U	I-S	I-U	O-S	O-U	I-S	I-U	O-S	O-U
ScrabbleGAN	1.327	1.309	1.193	1.184	26.95	27.48	30.62	33.86	0.996	0.986	0.997	0.986	0.052	0.041	0.117	0.119
ST-GAN	1.277	1.237	1.159	1.277	37.71	37.89	40.79	43.76	0.994	0.986	0.994	0.994	0.174	0.173	0.217	0.227
GANwriting	1.345	1.347	1.213	1.322	19.50	21.28	26.67	25.40	0.877	0.843	0.888	0.862	0.177	0.106	0.428	0.439
HWT	1.352	1.326	1.278	**1.369**	18.87	20.76	25.15	24.47	0.856	0.826	0.847	0.822	0.059	0.043	0.159	0.162
HiGAN	1.374	1.335	1.241	1.204	17.83	18.61	17.53	24.02	0.665	0.669	0.727	0.721	**0.004**	**0.003**	**0.061**	**0.061**
HiGAN+	**1.468**	**1.416**	**1.352**	1.296	**5.81**	**6.17**	**12.62**	**11.42**	**0.494**	**0.528**	**0.642**	**0.659**	0.008	0.005	0.092	0.086

Table 7. Comparison of Different GANs in Terms of Model Storage

Method	Size (MB)		
	Gen.	Enc.	Total
ScrabbleGAN	81.8	×	81.8
GANwriting	95.6	76.5	172.1
ST-GAN	**8.3**	127.5	135.8
HWT	80.7	50.6	131.3
HiGAN	*38.6	*20.5	*59.1
HiGAN+	15.0	**6.7**	**21.7**
Training Data			496.8

*Conference version.
In the table, "Gen." denotes the generator and "Enc." denotes the style encoder.

Our model sometimes fails in generating satisfactory handwriting images in the following two situations:

(1) HiGAN+ is difficult to synthesize realistic punctuation marks and digits. This is probably because that different characters and symbols in training data follow a long-tailed distribution, where the punctuation marks and digits are particularly rare in ground-truth samples. Therefore, HiGAN+ is good at synthesizing English characters rather than punctuation marks and digits.
(2) HiGAN+ may fail to generate extremely scribbled characters, while it prefers to generate neat and readable handwriting images. This is because the recognizer will penalize HiGAN+ during training if the model generates scribbled handwritten texts. This may be fixed by tuning the hyper-parameter λ_{ctc} of the text recognition loss $\mathcal{L}_{ctc}$ during training.

Overall, humans' handwriting can be very arbitrary and thus the proposed HiGAN+ essentially has limits for synthesizing meaningful handwriting images.

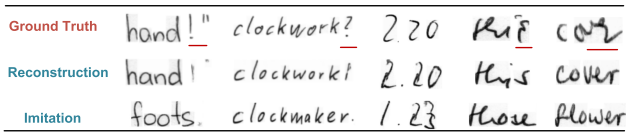


Fig. 16. Failure case analysis of HiGAN+.

4.9 Human Evaluation

Due to the subjective nature of images, we also conduct human evaluations (i.e., Turing tests) to verify the performance of different generative models for handwritten text synthesis. Specifically, we have conducted two user studies on a professional data platform Credamo with 100 randomly selected trustable participants who can recognize handwritten English texts.

4.9.1 User Plausibility Study. We first conduct a user plausibility study to test whether the synthetic images of HiGAN+ are actually indistinguishable from real ones by human judgements. In this study, we show each participant 50 random handwriting images (half genuine and half generated), where the participant can only view a single image at a time and then is asked whether the image is written by humans or artificially generated by machines. After ensuring the participant's reliability, there are 5,000 responses contributing to the final evaluation. As shown in Table 8, the study reveals that our generative model is clearly perceived as plausible.

4.9.2 User Preference Study. We also conduct a user preference study to justify whether HiGAN+ outperforms the competing GANs for handwritten text synthesis in terms of visual quality. In this study, we first randomly generate fake images of different GANs conditioned on the identified textual contents and calligraphic styles (where each GAN model generates one image at a time), and we repeat this procedure 25 times. After that, each participant is shown those images in a random order (side by side on the same screen) and then asked to choose the most preferred

Table 8. User Plausibility Study

Actual	Predicted		Overall Accuracy
	Real	Fake	
Genuine	0.3172	0.1828	0.5004
Generated	0.3168	0.1832	

Table 9. User Preference Study

Methods	Scrabble GAN	GAN writing	ST-GAN	HWT	HiGAN	HiGAN+
Prefers	0.0918	0.0542	0.1027	0.1686	0.2102	**0.3725**

one. In total, there are 2,500 responses contributing to the final evaluation. As shown in Table 9, our HiGAN+ obtains the majority of votes in all instances, which demonstrates the superiority of HiGAN+ over the competing GANs for handwritten text synthesis.

5 CONCLUSION

In this article, we have proposed a novel generative model HiGAN+ for HI based on disentangled representations. The proposed HiGAN+ can generate diverse and realistic handwritten texts conditioned on arbitrary textual contents and calligraphic styles (that are disentangled from reference images or randomly sampled from a prior normal distribution). Since conventional style transfer techniques based on pixel correspondences may be unsuitable for HI, we further introduce the contextual loss to significantly improve the style consistency of synthetic images. Moreover, to avoid many artifacts produced by existing GANs, we further refine the local details of synthetic handwriting images with an LPL. Lastly, we propose to reuse the early layers of the writer identifier for style encoding, thus deriving a more compact and effective architecture. Extensive experiments, including human evaluations, on the benchmark dataset demonstrate the superiority of HiGAN+ in terms of visual quality, scalability, compactness, and style transferability over the state-of-the-art GANs for handwritten text synthesis. It is worth noting that humans' handwriting is very arbitrary and thus HiGAN+ indeed has limits for synthesizing meaningful handwriting images. Nevertheless, it is interesting to teach machines/robotics to write texts as realistic as humans, which takes a closer step to high-level artificial intelligence. The source code of HiGAN+ is available at https://github.com/ganji15/HiGANplus.

REFERENCES

Eloi Alonso, Bastien Moysset, and Ronaldo Messina. 2019. Adversarial generation of handwritten text images conditioned on sequences. In *Proceedings of the International Conference on Document Analysis and Recognition*. 481–486.

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

Ankan Kumar Bhunia, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Mubarak Shah. 2021. Handwriting transformers. In *Proceedings of the International Conference on Computer Vision*. 1086–1094.

Mikolaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying MMD GANs. In *Proceedings of the International Conference on Learning Representations*.

Kaidi Cao, Jing Liao, and Lu Yuan. 2018. CariGANs: Unpaired photo-to-caricature translation. *ACM Transactions on Graphics* 37, 6 (2018), 244:1–244:14.

Junbum Cha, Sanghyuk Chun, Gayoung Lee, Bado Lee, Seonghyeon Kim, and Hwalsuk Lee. 2020. Few-shot compositional font generation with dual memory. In *Proceedings of the European Conference on Computer Vision*. 735–751.

Bo Chang, Qiong Zhang, Shenyi Pan, and Lili Meng. 2018. Generating handwritten Chinese characters using CycleGAN. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*. 199–207.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 2172–2180.

Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 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.

B. Davis, C. Tensmeyer, B. Price, C. Wigington, B. Morse, and R. Jain. 2020. Text and style conditioned GAN for generation of offline handwriting lines. arXiv:2009.00678. Retrieved from https://arxiv.org/abs/2009.00678.

Kingma P. Diederik and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference for Learning Representations*.

Sharon Fogel, Hadar Averbuch-Elor, Sarel Cohen, Shai Mazor, and Roee Litman. 2020. ScrabbleGAN: Semi-supervised varying length handwritten text generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4324–4333.

Ji Gan and Weiqiang Wang. 2021. HiGAN: Handwriting imitation conditioned on arbitrary-length texts and disentangled styles. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 7484–7492.

Yue Gao, Yuan Guo, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. 2019. Artistic glyph image synthesis via one-stage few-shot learning. *ACM Transactions on Graphics* 38, 6 (2019), 185:1–185:12.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*. 2672–2680.

Alex Graves. 2013. Generating sequences with recurrent neural networks. arXiv:1308.0850. Retrieved from https://arxiv.org/abs/1308.0850.

Alex Graves, Santiago Fernández, and Faustino Gomez. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*. 369–376.

David Ha and Douglas Eck. 2018. A neural representation of sketch drawings. In *Proceedings of the International Conference on Learning Representations*.

Tom S. F. Haines, Oisin Mac Aodha, and Gabriel J. Brostow. 2016. My text in your handwriting. *ACM Transactions on Graphics* 35, 3 (2016), 26:1–26:18.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 6626–6637.

Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. 2018. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision*. Vol. 11207, 179–196.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and locally consistent image completion. *ACM Transactions on Graphics* 36, 4 (2017), 107:1–107:14.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5967–5976.

Yue Jiang, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. 2019. SCFont: Structure-guided Chinese font generation via deep stacked networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, 4015–4022.

Lei Kang, Pau Riba, Yaxing Wang, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegasy. 2020. GANwriting: Content-conditioned generation of styled handwritten word images. In *Proceedings of the European Conference on Computer Vision*.

Lei Kang, Pau Riba, Marcal Rusinol, Alicia Fornes, and Mauricio Villegas. 2021. Content and style aware generation of text-line images for handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Early Access. DOI: https://doi.org/10.1109/TPAMI.2021.3122572

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. 2017. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the International Conference on Machine Learning*. 1857–1865.

Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. arXiv:1312.6114. Retrieved from https://arxiv.org/abs/1312.6114.

Atsunobu Kotani, Stefanie Tellex, and James Tompkin. 2020. Generating handwriting via decoupled style descriptors. In *Proceedings of the European Conference on Computer Vision*. Vol. 12357, 764–780.

Alex W. C. Lee, Jonathan Chung, and Marco Lee. 2021. GNHK: A dataset for English handwriting in the wild. In *Proceedings of the International Conference on Document Analysis and Recognition*. Josep Lladós, Daniel Lopresti, and Seiichi Uchida (Eds.), Springer, Cham, 399–412.

Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. 2018. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European Conference on Computer Vision*. Vol. 11205, 36–52.

Zhouchen Lin and Liang Wan. 2007. Style-preserving English handwriting synthesis. *Pattern Recognition* 40, 7 (2007), 2097–2109.

Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. 2019. Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1429–1437.

ZU-V. Marti and Horst Bunke. 2002. The IAM-Database: An English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition* 5, 1 (2002), 39–46.

Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. 2018. The contextual loss for image transformation with non-aligned data. In *Proceedings of the European Conference on Computer Vision*. Vol. 11218, 800–815.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. arXiv:1411.1784. Retrieved from https://arxiv.org/abs/1411.1784.

Song Park, Sanghyuk Chun, Junbum Cha, Bado Lee, and Hyunjung Shim. 2021. Few-shot font generation with localized style representations and factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 8024–8035.

Tiziano Portenier, Qiyang Hu, Attila Szabó, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. 2018. Faceshop: Deep sketch-based face image editing. *ACM Transactions on Graphics* 37, 4 (2018), 99:1–99:13.

Alec Radford, Luke Metz, and Soumith Chintala. 2013. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434. Retrieved from https://arxiv.org/abs/1511.06434.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 2226–2234.

Harm-de Vries, Florian Strub, Jeremie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C. Courville. 2017. Modulating early visual processing by language. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 6594–6604.

Yizhi Wang, Yue Gao, and Zhouhui Lian. 2020. Attribute2Font: Creating fonts you want from attributes. *ACM Transactions on Graphics* 39, 4, Article 69 (2020), 15 pages.

Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. 2019. Editing text in the wild. In *Proceedings of the ACM International Conference on Multimedia*. 1500–1508.

X. Y. Zhang, F. Yin, Y. M. Zhang, C. L. Liu, and Y. Bengio. 2018. Drawing and recognizing Chinese characters with recurrent neural network. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 4 (2018), 849–862.

Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. 2017. Toward multimodal image-to-image translation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 465–476.

Received July 2021; revised April 2022; accepted July 2022