Compositional Zero-Shot Artistic Font Synthesis

Xiang Li¹, Lei Wu²*, Changshuo Wang¹, Lei Meng²*, Xiangxu Meng²
School of Software, Shandong University
{202035260, 202115242}@mail.sdu.edu.cn, {i_lily, lmeng, mxx}@sdu.edu.cn

Abstract

Recently, many researchers have made remarkable achievements in the field of artistic font synthesis, with impressive glyph style and effect style in the results. However, due to less exploration in style disentanglement, it is difficult for existing methods to envision a kind of unseen style (glyph-effect) compositions of artistic font, and thus can only learn the seen style compositions. To solve this problem, we propose a novel compositional zero-shot artistic font synthesis gan (CAFS-GAN), which allows the synthesis of unseen style compositions by exploring the visual independence and joint compatibility of encoding semantics between glyph and effect. Specifically, we propose two contrast-based style encoders to achieve style disentanglement due to glyph and effect intertwining in the image. Meanwhile, to preserve more glyph and effect detail, we propose a generator based on hierarchical dual styles AdalIN to reorganize content-styles representations from structure to texture gradually. Extensive experiments demonstrate the superiority of our model in generating high-quality artistic font images with unseen style compositions against other state-of-the-art methods. The source code and data will be publicly available.

1 Introduction

Artistic fonts are frequently employed in signboards, posters, magazines, and web pages, playing an integral role in captivating and sustaining the audience’s attention. The compelling nature of these fonts lies in the fact that designers meticulously craft visually appealing and harmonious glyph and effect styles that suit the occasion and theme. In the course of design, the designers draw upon design theory and aesthetic factors to conceive various style elements, often requiring only a momentary mental picture. It is worth noting that if we can provide a deep learning model with enough glyph styles and effect styles as prior knowledge, whether the model can also design a kind of artistic font with unseen integrated style like humans.

In order to achieve the automatic synthesis of artistic font based on deep learning, some conventional methods [Azadi et al., 2018; Gao et al., 2019; Li et al., 2020a] focus on the integrated style (glyph-effect) transfer and generate an artistic font library with the existing style. These works treat the style of artistic fonts as a whole and generalize the learned integrated style to any character content. However, they ignore the independence and decoupling of styles, making these methods ineffective in scenarios where glyph and effect styles must be controlled separately. Therefore, the conventional methods cannot synthesize artistic fonts with unseen style (glyph-effect) compositions. There are also some recent works [Ge et al., 2021; Li et al., 2022a] that propose learning disentangled style representations and synthesizing content-glyph-effect controllable artistic font images. Unfortunately, these methods focus on the seen style compositions and must require a large amount of data paired with the three attributes

*Corresponding author
of content, glyph, and effect. Due to pixel-level supervision information, these methods inevitably focus on pixel-level relationship instead of creating the new style compositions, resulting in the generated images with a messy structure and unclear texture.

In this paper, we propose a novel and practical task, called compositional zero-shot artistic font synthesis (CAFS), which focuses on unseen style composition synthesis, see Figure 1. It aims to learn the compositionality of glyphs and effects from the training set and is tasked with generalizing to unseen style (glyph-effect) compositions on any character. To realize this task, we propose a new model, CAFS-GAN, from the perspective of style disentanglement and content-styles representations reorganization.

For the style disentanglement, we propose two contrast-based style encoders, glyph encoder and effect encoder, which implement glyph and effect disentanglement and precise style feature extraction. The key idea is that we introduce glyph style contrastive loss and effect style contrastive loss to learn the style commonalities and differences. For the content-styles representations reorganization, we propose an artistic font generator based on hierarchical dual styles AdaIN, which progressively feeds glyph and effect information to preserve more image details. The key idea is that the hierarchical dual styles AdaIN completes the composition of glyph and content in the high-dimensional AdaIN layer, and the composition of effect and content in the low-dimensional AdaIN layer. Moreover, to enable the model to synthesize artistic font images with controllable style attributes, we adopt the well-known GAN [Goodfellow et al., 2014] framework and introduce two multi-task discriminators, glyph discriminator and effect discriminator that constrain the style of the generated glyphs and effects, respectively. Finally, to comprehensively evaluate the generated results, we propose two evaluation metrics: glyph outline misalignment (GOLM) and effect perception error (EPE).

In summary, our contributions are as follows:

- We propose a novel compositional zero-shot artistic font synthesis gan (CAFS-GAN) to synthesize unseen style compositions for artistic font images. Meanwhile, our model supports the control of artistic font synthesis from three aspects (i.e., glyph, effect, and content).
- We propose two new evaluation metrics, called glyph outline misalignment (GOLM) and effect perception error (EPE), which enrich the evaluation methods from the unique attribute of the artistic font.
- Extensive experiments demonstrate the effectiveness and superiority of our model in synthesizing unseen style compositions in Chinese standard, creative, handwriting, calligraphy artistic fonts and English artistic fonts.

2 Related Work

2.1 Artistic Font Generation

Early artistic font generation approaches are based on the high regularity of the spatial distribution for effects. T-Effects [Yang et al., 2016] and DynTypo [Men et al., 2019] focus on texture and special effects for synthesizing complex and realistic artistic font images. TET-GAN [Yang et al., 2019a] and ShapeMatching-GAN [Yang et al., 2019b] establish the mapping between the original shape and the effect, using the CNN (Convolutional Neural Network) to realize the text effect transfer. Then, AGIS-Net [Gao et al., 2019] and FET-GAN [Li et al., 2020a] attempt the synchronous style transfer of glyphs and effects of arbitrary characters or symbols. Recently, DSE-Net [Li et al., 2022a] and GZS-Net [Ge et al., 2021] have conducted separate studies on the glyph structure and effects of artistic fonts. Although these methods separately encodes artistic font glyph and effects, they still have a significant data dependency on paired data. These models learn to synthesize artistic fonts by training on paired seen style combinations. Therefore, the optimization process for the model parameters is based on the pixel-level error between the generated and real images, which causes the model to focus excessively on pixel-level mapping relationships. This makes it difficult for the models to create new style combinations.

2.2 Disentangled Representation Learning

Disentangled representation learning aims to infer latent factors for a given object in the real world, where each latent factor is responsible for generating a semantic feature [Han et al., 2021; Yang et al., 2021; Saini et al., 2022]. Following VAE, [Higgins et al., 2017] introduces β-VAE to discover interpretable latent factor representations in a completely unsupervised manner. [Chen et al., 2018] improved β-VAE, and further proposed a principled classifier-free measure of disentanglement. Recently, a large amount of works [Zhang et al., 2018; Li et al., 2020b; Luo et al., 2022] have made great contributions to disentangled shape and texture, unfortunately, they are unable to generate novel combinations not witnessed during training.

2.3 Compositional Zero-Shot Learning

Compositional zero-shot learning stands at the intersection of compositionality and zero-shot learning and focuses on state and object relations. Compositionality [Naem et al., 2021] can loosely be defined as the ability to decompose an observation into its primitives. Zero-shot learning [Gao et al., 2018; Hong et al., 2022; Feng et al., 2022; Lin et al., 2022] aims at recognizing or generating novel classes that are not observed during training. Recently, [Yang et al., 2022] present a novel decomposable causal view that characterizes how compositional concepts are formed. [Karthik et al., 2022; Mancini et al., 2021] propose to address the problem of open-world compositional zero-shot learning. [Li et al., 2022b] propose a novel siamese contrastive embedding network to excavate discriminative prototypes of state and object.

In this paper, we propose a compositional zero-shot artistic font synthesis, and use the artistic font’s glyph and effect style as attribute primitives. More importantly, our method is the first to estimate the unseen style compositions, and uses the joint compatibility and differences between the two styles to synthesize and optimize the detailed characteristics of the image styles.
Figure 2: The overview of proposed CAFS-GAN. The encoding process of CAFS-GAN has three input channels (1) effect sample $s_x$, with effect attribute $x_i$, (2) glyph sample $s_y$, with glyph attribute $y_j$, and (3) content sample $s_z$, with content attribute $z_k$. For the style encoders, we additionally input positive samples $s^+_x$ and $s^+_y$ of style reference images. SSA integrates the style features of style samples and their positive samples from style encoders. The generator utilizes the hierarchical dual styles AdaIN architecture to reorganize the input content, effects, and glyph signals. The discriminator outputs a one-shot vector. The outputs of the discriminators in different channels indicate whether the generated image comes from the domain corresponding to this channel.

3 Method

3.1 Problem Define

Compositional zero-shot artistic font synthesis (CAFS) aims to predict an unseen style composition, namely to synthesize glyph-effect compositions that do not exist in the training set and map it to any character to obtain a complete artistic font library. Let us denote with $\mathcal{X} = \{x_i\}_{i=1}^{N_x}$ the set of effect attributes, with $\mathcal{Y} = \{y_j\}_{j=1}^{N_y}$ the set of glyph attributes, with $\mathcal{Z} = \{z_k\}_{k=1}^{N_z}$ the set of all possible compositions. $\mathcal{T} = \{T_i, C_i\}$ is a training set where $T_i$ is a character set seen during training ($\mathcal{T}_i \subseteq \mathcal{Z}$) and $C_i$ is a style compositions set seen during training ($\mathcal{C}_i \subseteq \mathcal{C}$). When the glyph and effect elements in $\mathcal{C}_i$ covers all elements in $\mathcal{X}$ and $\mathcal{Y}$, $T$ can be used to train the model $f: \{T_i, C_i\} \rightarrow \{\mathcal{Z}_i, \mathcal{C}_u\}$ synthesizing the artistic font images with unseen style combinations where $\mathcal{C}_u \subset \mathcal{C}$ denote the unseen style compositions and $\mathcal{C}_u \cup \mathcal{C}_u = \mathcal{C}$.

The difficulty of the CAFS task varies depending on the proportion of the $\mathcal{C}_i$. If the style compositions in $\mathcal{C}_i$ covers all compositions and $\mathcal{C}_u \equiv \emptyset$, the task definition is the same as the conventional artistic font generation task, where the model only needs to predict the seen style combination on arbitrary character content. In the case of $\mathcal{C}_i \subset \mathcal{C}$, since the model only learns jointly compatibility of encoding semantics between glyph and effect in seen style compositions, it is very challenging to predict unseen style combinations. It is worth noting that as the $\mathcal{C}_i$ shrinks, the training data can provide the model with fewer data on the joint compatibility relationship of glyph and effect. In this case, the shrink of composition information hinders the recognizability of glyph and effect, making it difficult for the model to predict unseen style combinations. Regarding this hypothesis, we verified it in Experiment 5.5.

3.2 Overview of CAFS-GAN

The CAFS-GAN consists of the following modules: two style encoders $E_x$ and $E_y$, two style similarity attention modules, a content encoder $E_z$, an artistic font generator $G$, and two style discriminators $D_{x}$ and $D_{y}$, as shown in Figure 2. First, $E_x$ and $E_y$ represent effect style encoder and glyph style encoder, respectively, which are used to disentangle and extract glyph and effect style features. At the end of the two style encoders, we add a style similarity attention (SSA) module, which uses the similarity of style attributes to enhance the model’s perception of various glyphs or effects. The structure details of $E_x$ and $E_y$ are similar to VGG11 [Simonyan and Zisserman, 2014]. Unlike $E_x$ and $E_y$, our $E_z$ adds several padding layers to increase the sampling times for the font strokes at the image’s edge. This operation protects the integrity of the character structure. In addition, since the content information of characters belongs to high-dimensional semantic information, we add resblocks at the end of the content encoder to retain more content information. Lastly, our
3.3 Contrast-Based Style Encoders

In the process of achieving the CAFS task, the style encoders need to provide the generator with disentangled glyph features and effect features. However, the actual situation is that the visual elements of effect, glyph, and content are entangled, and the commonly used data enhancement methods cannot eliminate or highlight a certain visual element. Therefore, we introduce a contrastive learning strategy to encourage encoders to identify deep similarities and differences between the two style attributes. Taking the pipeline of effect extraction as an example, we define $s_{x_i}$ and \{ $s_{x_1}, s_{x_2}, ..., s_{x_{n_x}}$ \} as the positive sample and negative sample set of the original input $s_{x_i}$, respectively. $N_x$ denotes the number of all kinds of effect styles, one of which is the effect of positive samples, and $N_x - 1$ is the number of all kinds of negative effects. The positive pair $(s_{x_i}, s_{x_i}^+)$ only shares the same effect, and the negative pair $(s_{x_i}, s_{x_i}^-)$ have different effects ($1 \leq r \leq N_x - 1$), as shown in Figure 3. At this time, we utilize the effect style contrastive loss to enhance the effect similarity between positive pairs and the dissimilarity between negative pairs:

$$ L_{sxy}^{E_s} = -\log \frac{\exp(f_{x_i} \cdot f_{x_i}^+ / \tau)}{\sum_{r=1}^{N_x-1} \exp(f_{x_i} \cdot f_{x_i}^- / \tau)} $$

(1)

where $f_{x_i}, f_{x_i}^+$, $f_{x_i}^-$ are effect features obtained by $s_{x_i}, s_{x_i}^+, s_{x_i}^-$ through $E_s$. Similarly, we also impose the glyph style contrastive loss $L_{sxy}^{E_g}$ to improve the glyph encoder. Furthermore, the total style contrastive loss can be defined as:

$$ L_{sxy} = L_{sxy}^{E_s} + L_{sxy}^{E_g} $$

(2)

3.4 Style Similarity Attention

To make full use of the style similarity between positive samples and original samples as auxiliary information for synthesizing disentangled style features, we introduce a style similarity attention module at the end of the style encoders. Specifically, we use the style features of positive samples as $K$ and $V$, and use the style features of original images as $Q$. Style similarity attention can be expressed as:

$$ SSA(Q, K, V) = \text{softmax}(\frac{f \cdot f^+ T}{\sigma})f^+ $$

(3)

where $f, f^+$ are style features from the original image and positive sample, and $\sigma$ factor follows Attention Mechanism [Vaswani et al., 2017] to prevent the magnitude of the dot product from growing extreme.

Overall, our proposed contrast-based style encoders encourage the encoders to have more robust style disentanglement ability. The SSA enhances the prominent glyph-effect characteristics by amplifying the specific style signal strength to obtain a pure glyph or effect representation.

3.5 Hierarchical Dual Styles AdaIN

Since neural networks are easier to retain abstract information in high-dimensional layers and easier to retain color information in low-dimensional layers [Gatys et al., 2016], we propose an artistic font generator based on hierarchical dual styles AdaIN. Specifically, we pass the disentangled glyph features and effect features through a fully connected layer (FC) to obtain high- and low-dimensional glyph style signals, respectively. Here, we input the glyph signal into the AdaIN layer [Huang and Belongie, 2017] of the generator and fuse the content information through high-dimensional connections, so that the generator can determine the overall outline and structural pattern in the early stage of generation. Furthermore, the effect signal is input to the generator through low-dimensional connections to render the color and texture details of the artistic font based on the established glyph. Formally, we use the style encoders and SSA to extract the effect feature $f_{x_i}$, and glyph features $f_{y_j}$, and input them to the fully connected layer. The fully connected layer aims to align $f_{x_i}$ and $f_{y_j}$ with the channel means and variances of the content inputs $f_{z_k}$, and to use $f_{x_i}$ and $f_{y_j}$ as the adaptive affine parameters of the AdaIN layer (i.e., $w$ and $b$). Ultimately, we achieve a progressive reorganization of the content with glyph and effect using hierarchical dual styles AdaIN:

$$ f_{z_k}^{l+1} = \begin{cases} w_{y_j} f_{x_i}^{l+1} + b_{y_j}, & l \leq h \\ w_{x_i} f_{y_j}^{l+1} + b_{x_i}, & l > h \end{cases} $$

(4)

where $l$ denotes the current layer number and $h$ denotes the threshold for dividing the high-dimensional AdaIN layers and the low-dimensional AdaIN layers.
3.6 Full objective

Our full objective functions can be summarized as follows:

$$\min_{G,E} \max_{D} \lambda_{sty} L_{sty} + \lambda_{adv} L_{adv}^x + \lambda_{adv} L_{adv}^y,$$  \hspace{1cm} (5)

where $\lambda_{sty}$ and $\lambda_{adv}$ are hyperparameters. The $L_{adv}^x$ and $L_{adv}^y$ denote two adversarial loss terms for the effect discriminator and glyph discriminator:

$$L_{adv}^x = \mathbb{E}[\log D_x(s_{x}, \cdot) + \log(1 - D_x(s_{x}, y, j, z))]$$ \hspace{1cm} (6)

$$L_{adv}^y = \mathbb{E}[\log D_y(s_{y}, \cdot) + \log(1 - D_y(s_{x}, y, j, z))],$$  \hspace{1cm} (7)

where $D_x(\cdot)$ and $D_y(\cdot)$ denote the logits from the domain-specific ($x_i$) effect discriminator and domain-specific ($y_j$) glyph discriminator. $s_{x}, s_{y}, j, z$ denote the generated artistic font image with three specific attributes.

4 Metrics

In order to better evaluate the generated glyphs and effects, we propose two kinds of new quantitative measures, GOLM for glyph and EPE for effect. Meanwhile, we also use three classic quantitative measures, such as $L_1$, SSIM, and FID.

**Glyp**h outline misalignment (GOLM). GOLM is used to evaluate whether the edge information of the generated artistic font is correct and complete. Firstly, we convert the images to their grayscale tic font images. EPE is used to evaluate whether the texture information of the generated image is accurate. Then, using the VGG19 [Simonyan and Zisserman, 2014] network to calculate the feature maps of the image in the deep layers, and then obtain the texture gram matrix [Gatys et al., 2016] through the inner product operation to represent the texture features. Then, EPE can be formulated as follows:

$$EPE = \frac{1}{n} \sum_{i=1}^{n} (G_i - G'_i)^2,$$ \hspace{1cm} (10)

where $n$ denotes the number of network layers involved in the calculation of feature maps, $G_i$ and $G'_i$ denote the gram matrices calculated in the $i$-layer network of the real image and the generated image.

5 Experiments

5.1 Datasets

**SSAF Dataset.** SSAF [Li et al., 2022a] contains a large number of high-quality Chinese and English artistic images, with annotations for their glyphs, effects, and content.

**Fonts Dataset.** Fonts [Ge et al., 2021] is a computer generated RGB font image dataset. It consists of 52 English letters with 5 independent attributes: letter identity, font size, font color, background color, and glyph.

5.2 Implementation Details

In our experiments, all images are resized to 128×128 pixels. The hyperparameters are set as: $\lambda_{adv} = 1.0$ and $\lambda_{sty} = 0.1$. In training, we set the batch size as 8 and train $10^5$ iterations for Chinese artistic font generation and $2 \times 10^4$ iterations for English. The learning rate is set to 0.0001, using Adam optimizer. Regarding the division of all possible style compositions, we set the proportion of the number of style compositions in $C_u$ to $C_l$ to be 1:8. In each category of artistic font, 775 Chinese characters and 22 uppercase English letters are used for training. 197 Chinese characters and 4 uppercase English letters are used for testing.

5.3 Comparison with SOTA Methods

**Quantitative comparison.** We compare three non-zero-shot methods, such as AGIS-Net [Gao et al., 2019], FET-GAN [Li et al., 2020a], and StarGANv2 [Choi et al., 2020]. The style (glyph-effect) compositions of the target artistic fonts synthesized by them are seen in the training. Meanwhile, we also compare two zero-shot methods, such as GZS-Net [Ge et al., 2021] and DSE-Net [Li et al., 2022a]. The style compositions they synthesized are unseen during training.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Disentangled Style</th>
<th>Training</th>
<th>$L_1$ loss ↓</th>
<th>FID ↓</th>
<th>SSIM ↑</th>
<th>GOLM ↓</th>
<th>EPE ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGIS-Net [Gao et al., 2019]</td>
<td>×</td>
<td>paired</td>
<td>0.2277</td>
<td>107.01</td>
<td>0.4313</td>
<td>81.025</td>
<td>4.3981</td>
</tr>
<tr>
<td>FET-GAN [Li et al., 2020a]</td>
<td>×</td>
<td>paired</td>
<td>0.2005</td>
<td>100.56</td>
<td>0.4474</td>
<td>68.820</td>
<td>7.5113</td>
</tr>
<tr>
<td>StarGANv2 [Choi et al., 2020]</td>
<td>×</td>
<td>unpaired</td>
<td>0.2997</td>
<td>72.24</td>
<td>0.3647</td>
<td>82.934</td>
<td>3.7708</td>
</tr>
</tbody>
</table>

Table 1: Quantitative comparison of the CAFS-GAN and the existing state-of-the-art methods.
Figure 4: Comparison with state-of-the-art methods. Manual results by human are shown in the last column as ground truth. Six rows of experimental results correspond to (1) Chinese artistic font with normal glyph. (2) Creative glyph. (3) Handwriting glyph. (4) Calligraphy glyph. (5) English artistic font with simple effect. (6) English artistic font with delicate effect.

Figure 5: Ablation study of CAFS-GAN. The Baseline includes three encoders and a generator without two style contrastive losses and SSA, and it receives two style vectors that have been spliced in the basic AdaIN layer. The setup of 5 groups of experiments: (A) adding $L_{Ex}$ to the Baseline, (B) incrementally adding $L_{Ey}$, (C) incrementally adding SSA, (D) replacing AdaIN with a reverse version of hierarchical dual styles AdaIN based on (C). (E) incrementally adding hierarchical dual styles AdaIN based on (C). The setup of experiment (E) denotes the full of CAFS-GAN.

5.4 Ablation Study

We conducted ablation study to validate the effectiveness of the components and loss functions of the model. The experimental results are depicted in Figure 5 and Table 2.

**Style contrastive losses.** The purpose of style contrastive losses is to disentangle the glyph and effect and improve the encoder’s ability to extract pure glyph and effect features. In Figure 5(A), after we add $L_{Ex}$, the dark red effect disappears obviously and the correct metal texture effect appears. After we simultaneously add $L_{Ex}$ and $L_{Ey}$, the glyph structure of (B) becomes more accurate than (A).

**Style similarity attention.** The SSA makes use of the style similarity between the positive and original samples to enhance the feature signal of the glyph and effect. We add SSA to the setup of experiment (B). In Figure 5(C), the stroke on the left side of this character has been significantly improved.

**Hierarchical dual styles AdaIN.** This structure helps the model to synthesize artistic fonts from structure to texture through hierarchically input to improve image details. The reverse version of this structure treats the glyph as low-dimensional information and the effect as high-dimensional information. We add the reverse version of hierarchical dual styles AdaIN to the setup of experiment (C). Figure 5 (C)(D) shows that the reverse version will lose a lot of effects and glyph details. Then, we add the right version of hierarchical dual styles AdaIN to the setup of experiment (C). Figure 5 (C)(E) shows the optimization of image details.
### Table 2: Quantitative evaluation of ablation study.

<table>
<thead>
<tr>
<th></th>
<th>L1 loss ↓</th>
<th>FID ↓</th>
<th>SSIM ↑</th>
<th>GOLM ↓</th>
<th>EPE ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>0.2750</td>
<td>261.08</td>
<td>0.3039</td>
<td>189.29</td>
<td>2.6751</td>
</tr>
<tr>
<td>(A)</td>
<td>0.2852</td>
<td>257.73</td>
<td>0.2653</td>
<td>187.51</td>
<td>3.9582</td>
</tr>
<tr>
<td>(B)</td>
<td>0.2336</td>
<td>201.66</td>
<td>0.3345</td>
<td>183.83</td>
<td>2.0330</td>
</tr>
<tr>
<td>(C)</td>
<td>0.2290</td>
<td>201.66</td>
<td>0.3345</td>
<td>183.83</td>
<td>2.0330</td>
</tr>
<tr>
<td>(D)</td>
<td>0.2452</td>
<td>262.31</td>
<td>0.3099</td>
<td>185.10</td>
<td>1.6954</td>
</tr>
<tr>
<td>(E)</td>
<td>0.2251</td>
<td>179.61</td>
<td>0.3520</td>
<td>179.25</td>
<td>1.0767</td>
</tr>
</tbody>
</table>

### 5.5 Proportion of the Seen Style Compositions

We also discussed the influence of the proportion of seen style composition \( C_t \) to all possible style compositions \( C \) on the experimental results. We use six different training sets to train CAFS-GAN, each containing the same three effects and three glyphs, but their number of compositions is different. The ratios of style combinations of \( C_t \) to \( C \) are set to 4/9, 5/9, 6/9, 7/9, 8/9, and 9/9. As shown in Figure 6, with the proportion increase, the model’s performance presents an overall improved state. Therefore, we concluded that sufficient glyph-effect joint compatibility relationship will improve the model’s ability to understand the artistic font’s attributes and help the model synthesize unseen style compositions.

### 5.6 Visualization

In order to further demonstrate the style disentanglement capability of the \( E_x \) and \( E_y \) and the ability to recombine content and styles of the generator, we visualize the attention maps generated by style encoders and feature maps generated by the generator. In Figure 7(a), we feed three different effects of the artistic font images to \( 1 \) and \( 1 \), and the texture part of these images got a lot of attention. In Figure 7(b), the glyph encoder tends to focus on local areas of artistic fonts, which are the unique characteristics of the glyphs, such as curves and corners. In Figure 7(c), the structure of feature maps of fonts are changed firstly (e.g., the lines become clear, and the corners become apparent). Then, there is more pixel filling inside the feature maps of the font. After that, the texture is rendered.

### 5.7 Style Interpolation

We further demonstrate the flexibility of CAFS-GAN through glyph style interpolation and effect style interpolation. In CAFS-GAN, we can explicitly control the weighting between different glyph or effect representations and decode the integrated representation back to the image space, obtaining the new mixed attributes, see Figure 8. This is meaningful to the diversification of artistic fonts.

### 5.8 Conclusion

In this paper, we propose a new task called compositional zero-shot artistic font synthesis (CAFS), which allows synthesizing arbitrary character’s artistic font image with unseen style compositions. To achieve this task, we propose the CAFS-GAN model, focusing on style disentanglement of glyph and effect, and hierarchical reorganization of content and styles representations. We also propose two evaluation metrics for a more comprehensive evaluation of artistic font images: glyph outline misalignment and effect perception error. Extensive experiments demonstrate the effectiveness of our model’s multi-attributes control and the superiority of generation quality over existing methods.
References


[Han et al., 2021] Yuxuan Han, Jiaolong Yang, and Ying Fu. Disentangled face attribute editing via instance-aware latent space search. In IJCAI, pages 715–721, 2021.


