Compositional Zero-Shot Artistic Font Synthesis

Xiang Li[1], Lei Wu[2*], Changshuo Wang[1], Lei Meng[2*], Xiangxu Meng[2]
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 AdaIN 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.

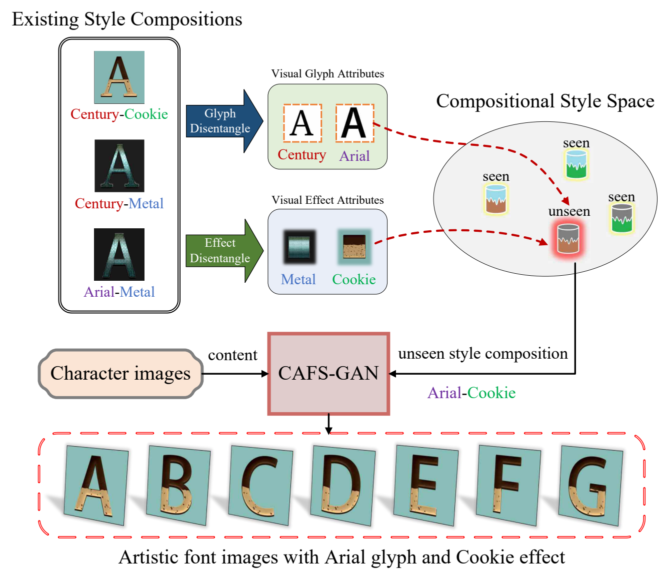


Figure 1: We aim to build an artistic font synthesis model for synthesizing unseen style compositions (e.g., Arial-Cookie) by training with some seen concepts, such as Century-Cookie, Century-Metal, and Arial-Metal.

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 [Naeem *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.

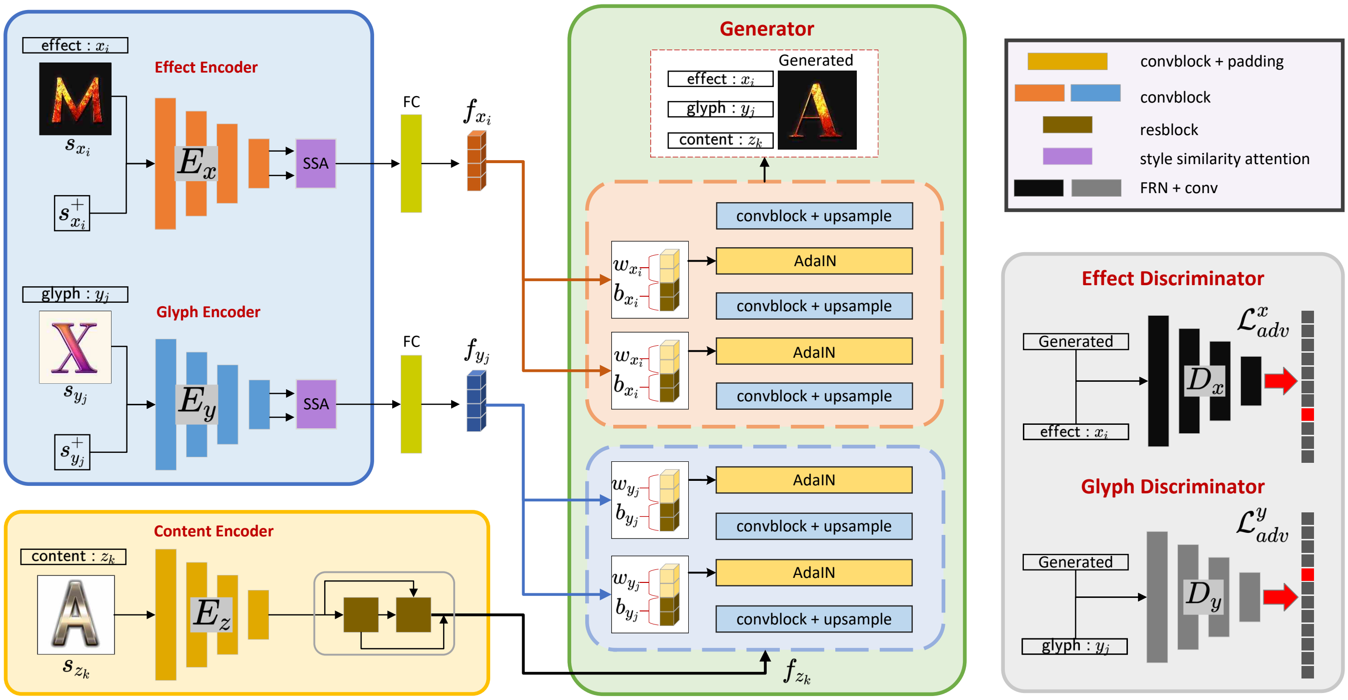


Figure 2: The overview of proposed CAFS-GAN. The encoding process of CAFS-GAN has three input channels (1) effect sample s_{x_i} with effect attribute x_i, (2) glyph sample s_{y_j} with glyph attribute y_j, and (3) content sample s_{z_k} with content attribute z_k. For the style encoders, we additionally input positive samples $s^+_{x_i}$ and $s^+_{y_j}$ 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 characters, and with $\mathcal{C} = \mathcal{X} \times \mathcal{Y}$ the set of all their possible compositions. $\mathcal{T} = \{\mathcal{Z}_t, \mathcal{C}_t\}$ is a training set where $\mathcal{Z}_t$ is a character set seen during training ($\mathcal{Z}_t \subseteq \mathcal{Z}$) and $\mathcal{C}_t$ is a style compositions set seen during training ($\mathcal{C}_t \subseteq \mathcal{C}$). When the glyph and effect elements in $\mathcal{C}_t$ covers all elements in $\mathcal{X}$ and $\mathcal{Y}$, $\mathcal{T}$ can be used to train the model $f : \{\mathcal{Z}_t, \mathcal{C}_t\} \rightarrow \{\mathcal{Z}_t, \mathcal{C}_u\}$ synthesizing the artistic font images with unseen style combinations where $\mathcal{C}_u \subset \mathcal{C}$ denote the unseen style compositions and $\mathcal{C}_t \cup \mathcal{C}_u = \mathcal{C}$.

The difficulty of the CAFS task varies depending on the proportion of the $\mathcal{C}_t$. If the style compositions in $\mathcal{C}_t$ 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 $\mathcal{C}_t \subset \mathcal{C}$, 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 $\mathcal{C}_t$ 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

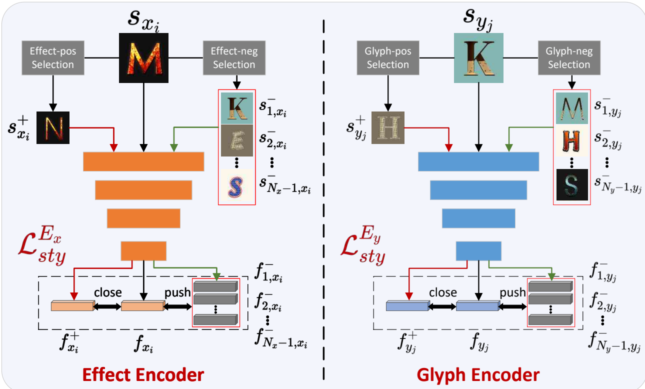


Figure 3: Two contrast-based style encoders. s_{x_i} and s_{y_j} represent effect and glyph samples. $s^+_{x_i}$ and $s^+_{y_j}$ have the same visual attributes as s_{x_i} and s_{y_j} in the corresponding attributes, respectively, and vice versa for effect and glyph negative samples.

D_x and D_y are two multi-task discriminators consisting of FRN (Filter Response Normalization) [Singh and Krishnan, 2020] and convolutional layer, which consists of multiple output branches. Each branch learns a binary classification determining whether an artistic font has real glyph style or real effect style.

In the next sections, we will look at the two aspects: style (glyph-effect) disentanglement (in sections 3.3 and 3.4) and content-styles representations reorganization (in section 3.5).

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 [He *et al.*, 2020] 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 $\{s^-_{1,x_i}, s^-_{2,x_i}, ..., s^-_{N_x-1,x_i}\}$ 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^-_{r,x_i}) have different effects ($1 \le r \le N_x - 1$), 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:

$$\mathcal{L}^{E_x}_{sty} = -\log \frac{\exp(f_{x_i} \cdot f^+_{x_i}/\tau)}{\sum_{r=1}^{N_x-1} \exp(f_{x_i} \cdot f^-_{r,x_i}/\tau)}, \quad (1)$$

where $f_{x_i}, f^+_{x_i}, f^-_{r,x_i}$ are effect features obtained by s_x, s^+_x, s^-_{r,x_i} through E_x. Similarly, we also impose the glyph style contrastive loss $\mathcal{L}^{E_y}_{sty}$ to improve the glyph encoder. Furthermore, the total style contrastive loss can be defined as:

$$\mathcal{L}_{sty} = \mathcal{L}^{E_x}_{sty} + \mathcal{L}^{E_y}_{sty}. \quad (2)$$

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:

$$\mathrm{SSA}(Q, K, V) = softmax(\frac{f \cdot {f^+}^T}{\sigma}) f^+, \quad (3)$$

where f, f^+ are style features from the original image and positive sample, and σ 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:

$$f^{l+1}_{z_k} = \begin{cases} w_{y_j}(\frac{f^l_{z_k} - \mu}{\sigma}) + b_{y_j}, & l \le h \\ w_{x_i}(\frac{f^l_{z_k} - \mu}{\sigma}) + b_{x_i}, & l > h \end{cases} \quad (4)$$

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.

Methods	Disentangled Style	Training	L_1 loss ↓	FID ↓	SSIM ↑	GOLM ↓	EPE ↓
Non-zero-shot methods for synthesizing seen style compositions							
AGIS-Net [Gao *et al.*, 2019]	×	paired	0.2277	107.01	0.4313	81.025	4.3981
FET-GAN [Li *et al.*, 2020a]	×	paired	0.2005	100.56	0.4474	68.820	7.5113
StarGANv2 [Choi *et al.*, 2020]	×	unpaired	0.2997	72.24	0.3647	82.934	3.7708
Zero-shot methods for synthesizing unseen style compositions							
GZS-Net [Ge *et al.*, 2021]	✓	paired	0.2460	140.35	0.3648	87.328	7.2335
DSE-Net [Li *et al.*, 2022a]	✓	paired	0.1754	72.19	0.4428	83.345	3.7332
Ours	✓	unpaired	**0.1271**	**64.79**	**0.5883**	**73.225**	**3.0734**

Table 1: Quantitative comparison of the CAFS-GAN and the existing state-of-the-art methods.

3.6 Full objective

Our full objective functions can be summarized as follows:

$$\min_{G,E} \max_{D} \lambda_{sty}\mathcal{L}_{sty} + \lambda_{adv}\mathcal{L}^{x}_{adv} + \lambda_{adv}\mathcal{L}^{x}_{adv}, \tag{5}$$

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

$$\mathcal{L}^{x}_{adv} = \mathbb{E}[\log D_{x_i}(s_{x_i}) + \log(1 - D_{x_i}(s_{x_i,y_j,z_k}))], \tag{6}$$

$$\mathcal{L}^{y}_{adv} = \mathbb{E}[\log D_{y_j}(s_{y_i}) + \log(1 - D_{y_j}(s_{x_i,y_j,z_k}))], \tag{7}$$

where $D_{x_i}(\cdot)$ and $D_{y_j}(\cdot)$ denote the logits from the domain-specific (x_i) effect discriminator and domain-specific (y_j) glyph discriminator. s_{x_i,y_j,z_k} 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.

Glyph 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 I to its grayscale I_{gray}, and calculate horizontal and vertical directions gradients using the Sobel operator. By summing the root mean square of the gradients in the two directions, we can get the final gradient of each pixel. The formula for GOLM is as follows:

$$GOLM = \left| I_{edge} - I'_{edge} \right|, \tag{8}$$

$$I_{edge} = \sqrt{(A \cdot I_{gray})^2 + (B \cdot I_{gray})^2}, \tag{9}$$

where I_{edge} and I'_{edge} denote the edge image of the real image and generated image. A and B denote horizontal and vertical Sobel matrixs.

Effect perception error (EPE). The visual communication of effect is often presented in the form of texture in artistic font images. EPE is used to evaluate whether the texture information of the generated image is accurate. First, we use 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:

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

where n denotes the number of network layers involved in the calculation of feature maps, $\mathcal{G}_i$ and $\mathcal{G}'_i$ denote the gram matrixs 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 2×10^4 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 $\mathcal{C}_u$ to $\mathcal{C}_t$ 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. In Table 1,

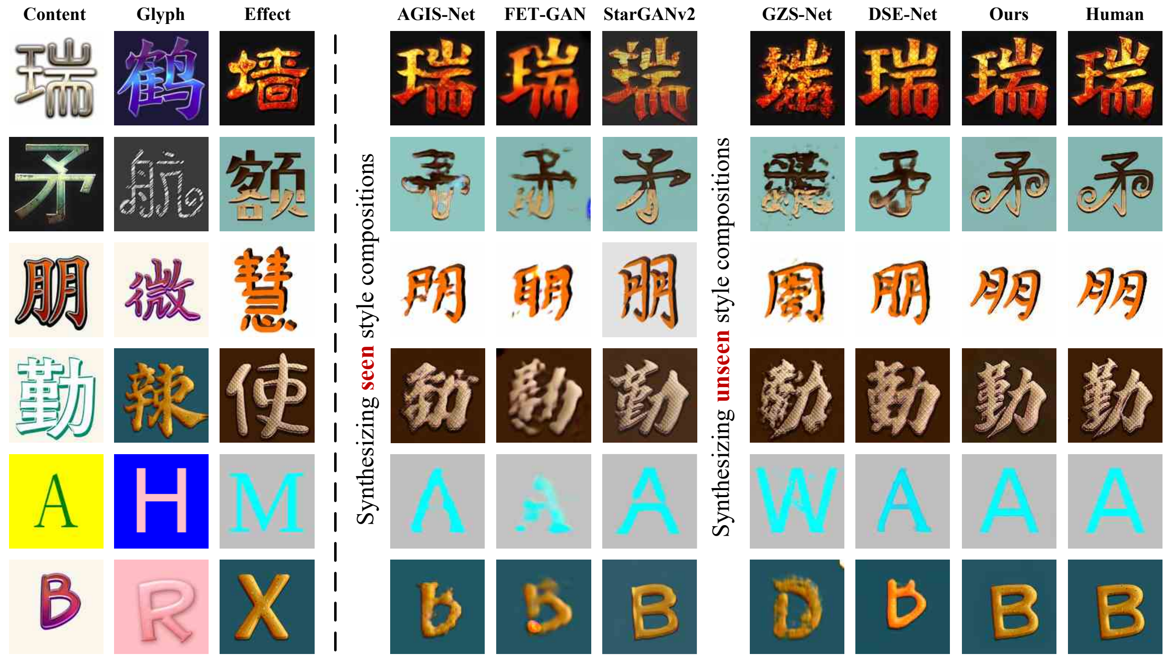


Figure 4: Comparison with state-of-the-art methods. Manual results by human are shown in the last column as ground truth. Six rows of experimental results correspond to (1) Chinese artistic font with normal glyph. (2) Creative glyph. (3) Handwriting glyph. (4) Calligraphy glyph. (5) English artistic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

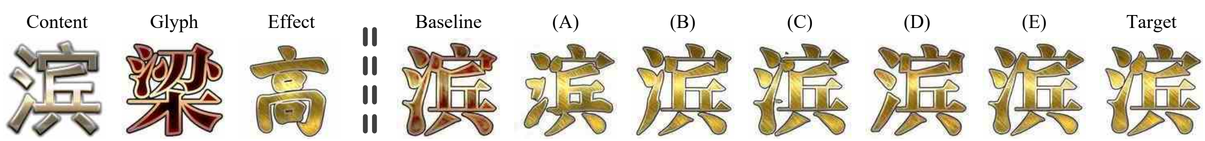


Figure 5: Ablation study of CAFS-GAN. The Baseline includes three encoders and a generator without two style contrastive losses and SSA, and it receives two style vectors that have been spliced in the basic AdaIN layer. The setup of 5 groups of experiments: (A) adding $\mathcal{L}_{sty}^{E_x}$ to the Baseline, (B) incrementally adding $\mathcal{L}_{sty}^{E_y}$, (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.

the CAFS-GAN proposed by us has achieved apparent advantages in synthesizing unseen style compositions. Moreover, the synthesized results by CAFS-GAN are also ahead of the conventional artistic font synthesis methods in five metrics.

Qualitative comparison. In Figure 4, our method has generated photo-realistic glyph and effect style and is superior to other methods. We can easily observe that some methods work well in the normal glyph, but their performance in creative, handwriting, and calligraphy drops sharply. For English, some methods are difficult to generate the correct glyph and effect (e.g., DSE-Net), and the othes are difficult to generate the correct character content (e.g., GZS-Net).

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 $\mathcal{L}_{sty}^{E_x}$, the dark red effect disappears obviously and the correct metal texture effect appears. After we simultaneously add $\mathcal{L}_{sty}^{E_x}$ and $\mathcal{L}_{sty}^{E_y}$, 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.

	L_1 loss ↓	FID ↓	SSIM ↑	GOLM ↓	EPE ↓
Baseline	0.2750	261.08	0.3039	189.29	2.6751
(A)	0.2852	257.73	0.2653	187.51	3.9582
(B)	0.2336	201.66	0.3345	183.83	2.0330
(C)	0.2290	**178.07**	0.3333	182.81	1.2076
(D)	0.2452	262.31	0.3099	185.10	1.6954
(E)	**0.2251**	179.61	**0.3520**	**179.25**	**1.0767**

Table 2: Quantitative evaluation of ablation study.

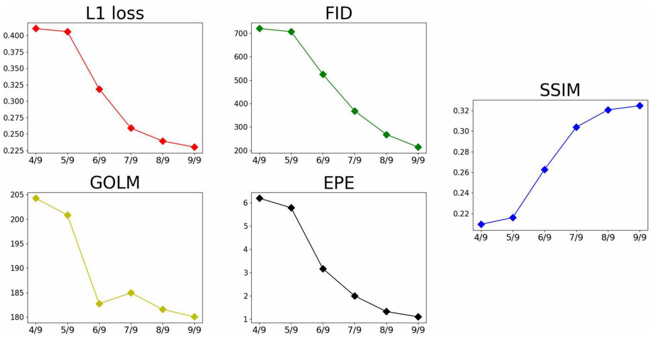


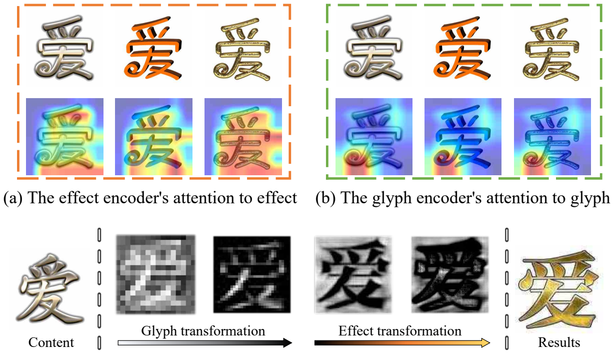
Figure 6: Influence of the proportion of seen style compositions. The x-coordinate represents the proportion of $\mathcal{C}_t$ to $\mathcal{C}$, and the y-coordinate represents the value of each metric.

5.5 Proportion of the Seen Style Compositions

We also discussed the influence of the proportion of seen style composition $\mathcal{C}_t$ to all possible style compositions $\mathcal{C}$ 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 $\mathcal{C}_t$ to $\mathcal{C}$ 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.



(a) The effect encoder's attention to effect (b) The glyph encoder's attention to glyph

(c) The evolution of glyph and effect in generator

Figure 7: The visualization of the style attention maps and generated feature maps. (a)(b) We show the original images (in the first row) and their attention maps of glyph and effect (in the second row) in two style encoders. (c) We show how the generator adjusts the structure and then renders the effect.

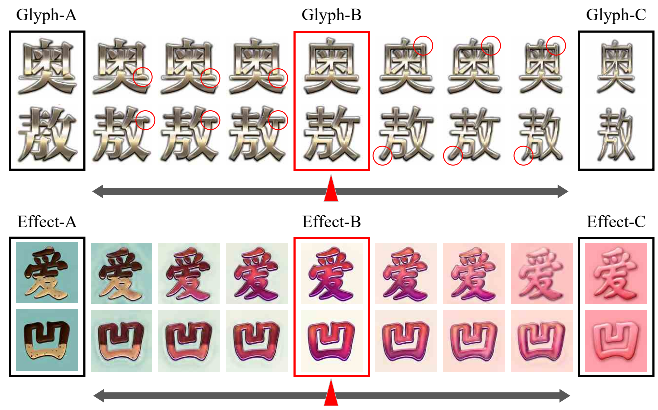


Figure 8: Glyph style interpolation and effect style interpolation.

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

[Azadi *et al.*, 2018] Samaneh Azadi, Matthew Fisher, Vladimir G Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content gan for few-shot font style transfer. In *CVPR*, pages 7564–7573, 2018.

[Chen *et al.*, 2018] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in vaes. In *NIPS*, pages 2615–2625, 2018.

[Choi *et al.*, 2020] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, pages 8188–8197, 2020.

[Feng *et al.*, 2022] Yaogong Feng, Xiaowen Huang, Pengbo Yang, Jian Yu, and Jitao Sang. Non-generative generalized zero-shot learning via task-correlated disentanglement and controllable samples synthesis. In *CVPR*, pages 9346–9355, 2022.

[Gao *et al.*, 2018] Rui Gao, Xingsong Hou, Jie Qin, Li Liu, Fan Zhu, and Zhao Zhang. A joint generative model for zero-shot learning. In *ECCV*, pages 631–646, 2018.

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

[Gatys *et al.*, 2016] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016.

[Ge *et al.*, 2021] Yunhao Ge, Sami Abu-El-Haija, Gan Xin, and Laurent Itti. Zero-shot synthesis with group-supervised learning. In *ICLR*, 2021.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NIPS*, page 2672–2680, 2014.

[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.

[He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.

[Higgins *et al.*, 2017] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[Hong *et al.*, 2022] Ziming Hong, Shiming Chen, Guo-Sen Xie, Wenhan Yang, Jian Zhao, Yuanjie Shao, Qinmu Peng, and Xinge You. Semantic compression embedding for generative zero-shot learning. In *IJCAI*, pages 956–963, 2022.

[Huang and Belongie, 2017] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1501–1510, 2017.

[Karthik *et al.*, 2022] Shyamgopal Karthik, Massimiliano Mancini, and Zeynep Akata. Kg-sp: Knowledge guided simple primitives for open world compositional zero-shot learning. In *CVPR*, pages 9336–9345, 2022.

[Li *et al.*, 2020a] Wei Li, Yongxing He, Yanwei Qi, Zejian Li, and Yongchuan Tang. Fet-gan: Font and effect transfer via k-shot adaptive instance normalization. In *AAAI*, pages 1717–1724, 2020.

[Li *et al.*, 2020b] Yuheng Li, Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Mixnmatch: Multifactor disentanglement and encoding for conditional image generation. In *CVPR*, pages 8039–8048, 2020.

[Li *et al.*, 2022a] Xiang Li, Lei Wu, Xu Chen, Lei Meng, and Xiangxu Meng. Dse-net: Artistic font image synthesis via disentangled style encoding. In *ICME*, pages 1–6, 2022.

[Li *et al.*, 2022b] Xiangyu Li, Xu Yang, Kun Wei, Cheng Deng, and Muli Yang. Siamese contrastive embedding network for compositional zero-shot learning. In *CVPR*, pages 9326–9335, 2022.

[Lin *et al.*, 2022] Chung-Ching Lin, Kevin Lin, Lijuan Wang, Zicheng Liu, and Linjie Li. Cross-modal representation learning for zero-shot action recognition. In *CVPR*, pages 19978–19988, 2022.

[Luo *et al.*, 2022] Canjie Luo, Lianwen Jin, and Jingdong Chen. Siman: Exploring self-supervised representation learning of scene text via similarity-aware normalization. In *CVPR*, pages 1039–1048, 2022.

[Mancini *et al.*, 2021] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Open world compositional zero-shot learning. In *CVPR*, pages 5222–5230, 2021.

[Men *et al.*, 2019] Yifang Men, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. Dyntypo: Example-based dynamic text effects transfer. In *CVPR*, pages 5870–5879, 2019.

[Naeem *et al.*, 2021] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *CVPR*, pages 953–962, 2021.

[Saini *et al.*, 2022] Nirat Saini, Khoi Pham, and Abhinav Shrivastava. Disentangling visual embeddings for attributes and objects. In *CVPR*, pages 13658–13667, 2022.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[Singh and Krishnan, 2020] Saurabh Singh and Shankar Krishnan. Filter response normalization layer: Eliminating batch dependence in the training of deep neural networks. In *CVPR*, pages 11237–11246, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, pages 5998–6008, 2017.

[Yang *et al.*, 2016] Shuai Yang, Jiaying Liu, Zhouhui Lian, and Zongming Guo. Awesome typography: Statistics-based text effects transfer. *CoRR*, abs/1611.09026, 2016.

[Yang *et al.*, 2019a] Shuai Yang, Jiaying Liu, Wenjing Wang, and Zongming Guo. Tet-gan: Text effects transfer via stylization and destylization. In *AAAI*, pages 1238–1245, 2019.

[Yang *et al.*, 2019b] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. Controllable artistic text style transfer via shape-matching gan. In *CVPR*, pages 4442–4451, 2019.

[Yang *et al.*, 2021] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *CVPR*, pages 9593–9602, 2021.

[Yang *et al.*, 2022] Muli Yang, Chenghao Xu, Aming Wu, and Cheng Deng. A decomposable causal view of compositional zero-shot learning. *IEEE Transactions on Multimedia*, pages 1–11, 2022.

[Zhang *et al.*, 2018] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *CVPR*, pages 2694–2703, 2018.