Frequency Domain Disentanglement for Arbitrary Neural Style Transfer

Dongyang Li, Hao Luo*, Pichao Wang, Zhibin Wang*, Shang Liu, Fan Wang†

Alibaba Group

{yingtian.ldy, michuan.lh, pichao.wang, zhibin.waz, liushang.ls, fan.w}@alibaba-inc.com

Abstract

Arbitrary neural style transfer has been a popular research topic due to its rich application scenarios. Effective disentanglement of content and style is the critical factor for synthesizing an image with arbitrary style. The existing methods focus on disentangling feature representations of content and style in the spatial domain where the content and style components are innately entangled and difficult to be disentangled clearly. Therefore, these methods always suffer from low-quality results because of the sub-optimal disentanglement. To address such a challenge, this paper proposes the frequency mixer (FreMixer) module that disentangles and re-entangles the frequency spectrum of content and style components in the frequency domain. Since content and style components have different frequency-domain characteristics (frequency bands and frequency patterns), the FreMixer could well disentangle these two components. Based on the FreMixer module, we design a novel Frequency Domain Disentanglement (FDD) framework for arbitrary neural style transfer. Qualitative and quantitative experiments verify that the proposed method can render better stylized results compared to the state-of-the-art methods.

Introduction

Style transfer, which aims to generate an image I_{cs} applying the style s of image I_s to the content c of image I_c (Efros and Freeman 2001; Drori, Cohen-Or, and Yeshurun 2003; Frigo et al. 2016; Elad and Milanfar 2017), has attracted significant research attention. Recent works have shown that the key point of synthesizing a stylized image is how to disentangle the *content* and *style* from the image. According to the different constraints, mainstream approaches for learning content-style disentanglement include loss constraints (Gatys, Ecker, and Bethge 2016; Johnson, Alahi, and Fei-Fei 2016), global mean and variance constraints (AdaIN) (Huang and Belongie 2017) and adversarial constraints (Kotovenko et al. 2019; Chen et al. 2021). Recently, some works (Park and Lee 2019; Yao et al. 2019; Wu et al. 2021; Liu et al. 2021; Deng et al. 2022) introduced the spatial attention mechanism to learn point-wise weights for content and style, respectively. Although these methods have significantly improved the performance in neural style transfer, they still suffer from problems such as local distortions, weak generalization and inflexibility because of the innate entanglement of content and style in the spatial domain. Specifically, the content and style components closely coexist in the image patch and are difficult to disentangle, which leads to the effects of loss of detailed contents or broken structure in the stylized images.

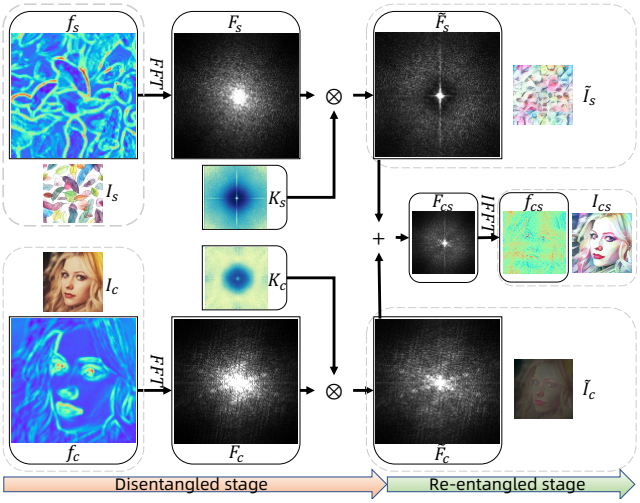

Figure 1: The disentanglement and re-entanglement of content and style in the frequency domain. The subscripts c and s represent that current node is from the content image I_c and style image I_s, respectively. f denotes the feature map of the convolutional layer. $F/\widetilde{F}$ stands for the original/disentangled frequency spectrum. K is the global frequency kernel. All f, F, $\widetilde{F}$ and K have the same spatial size.

The image/feature map can be transformed into the frequency domain where the low-frequency bands describe the smoothly changing parts and the high-frequency bands are related to the rapidly changing parts (Chen et al. 2019), and we call this property as **Frequency Separable Property (FSP)**. It is also well known that in signal processing, some signal characteristics are easy to analyze in the frequency domain but challenging to explore in the spatial domain. Recently, several works try to apply **FSP** to solve

* Corresponding author, † Project lead.

problems in different fields. (Chen et al. 2019) proposed to store and process the feature maps that change slowly at lower spatial resolution to reduce the computation cost. In de-raining fields, there is a priori knowledge following **FSP** that the object structure information is always homologous with the high-frequency bands while the rain streaks are related to the low-frequency bands in the image (Perona and Malik 1990; Fu et al. 2011, 2017). (Yang and Soatto 2020) changed the image's appearance without altering semantic information to perform the domain alignment via swapping the low-frequency between two images. Frequency Domain Image Translation (FDIT) (Cai et al. 2021) preserves the identity of the reconstruction image via the high-frequency consistency. All of these works reflect that analyzing the image feature in frequency domain is more effective than in the spatial domain, which inspires us to separate the content and style components for arbitrary neural style transfer in the frequency domain.

In this paper, we propose the Frequency Domain Disentanglement (FDD) approach for arbitrary neural style transfer. As shown in Figure 1, the key point is to transform the spatial features f_* encoded by the pre-trained VGG-19 into the frequency domain via the fast Fourier transform algorithm (FFT) (Cooley and Tukey 1965), where the content and style components can be naturally disentangled and re-entangled because of their different frequency-domain characteristics. In this paper, we propose a novel module called frequency mixer (FreMixer), which is conceptually simple and computationally efficient. FreMixer consists of the global frequency kernels to disentangle and re-entangle the content and style components. The basic idea behind the FreMixer module is to learn the different frequency patterns for content and style. By stacking FreMixer modules, FDD can learn different frequency kernels K_* to disentangle detail content information and unique style. Then, the reintegrated frequency spectrum is transformed into the spatial domain via the inverse fast Fourier transform (IFFT). Our work demonstrates that the disentanglement of content and style in the frequency domain is more straightforward than in the spatial domain.

We conduct extensive experiments to verify the effectiveness of the proposed method. Different from previous works that mainly present the standard stylization, we conduct experiments with additional settings including *progressive and extrapolative stylization*, (and *cross-datasets style transfer*, *progressive content interpolation*, *bidirectional stylization* in the Appendix) to show that our method can successfully disentangle content and style in the frequency domain. For better understanding of our method, we also visualize the learnable frequency-domain kernels to show different frequency-domain characteristics between content and style components. Quantitative and qualitative experimental results on different datasets demonstrate that our method achieves significant performance improvements compared to the SOTA methods. To our best knowledge, we are the first to explore content-style disentanglement for the arbitrary neural style transfer from a frequency perspective. We believe our work will provide new insights to the community.

Related Work

Arbitrary Neural Style Transfer

Before applying the convolutional neural networks (CNNs) to style transfer, the researchers has explored a related field called image-based artistic rendering (IB-AR) (Kyprianidis et al. 2012). Due to the limitations of IB-AR, such as flexibility, style diversity and the disability of disentangling the content structure and style texture, the arbitrary neural style transfer (ANST) (Jing et al. 2019) was proposed to tackle these problems. To our best knowledge, the early work (Gatys, Ecker, and Bethge 2016) is the first work that uses CNN to synthesize the image with the different styles. The work demonstrates that the response of feature maps from a pre-trained CNN encoder can represent the content information, and the statistical distributions can represent the style pattern. Huang *et al.* proposed a novel adaptive instance normalization (AdaIN) layer to produce the stylized features by aligning the mean and variance between the content features and style features in a global manner (Huang and Belongie 2017). (Chen et al. 2021) introduced contrastive loss and adversarial loss to force the model to learn style-to-style relations and human-aware style information. Recently, attention mechanism shows the great strength in style transfer. For example, (Park and Lee 2019) proposed a style-attentional network (SANet) to transfer the style according to the spatial distribution of the content image. (Liu et al. 2021) proposed adaptive attention normalization (AdaAttN) which integrates the advantages of AdaIN and SANet to adaptively perform attentive normalization on a per-point basis. Some works (Wu et al. 2021; Deng et al. 2022) introduce cross-attention transformers to stylize the content feature sequence according to the style feature sequence. However, all of the above methods conduct style transfer in the spatial domain, where the content and style cannot be disentangled well due to the worse separation property of the spatial domain. In contrast, we propose to explore content-style disentanglement in the frequency domain, as it has better separable property.

Applications of Frequency Separable Property

FSP is a commonly used technique in different fields. (Chen et al. 2019) clam that the convolutional feature maps are mixture of information at different frequencies and propose to factorize the feature maps into two categories with different frequency bands. The lower frequency bands are processed at lower spatial-resolution CNN layer to reduce the computation complexity. In de-raining field, (Fu et al. 2017) employ a priori image domain knowledge via focusing on the high-frequency information during the training stage, which forces the model to learn the structure information of the rain. In the domain adaptation segmentation task, in order to generate the training data similar with the target domain, (Yang and Soatto 2020) propose the spectral transfer to map a source image to a target domain by simply swapping the low-frequency component between source image and target image. In the image-to-image translation, due to the absence of preserving the identity of the source domain, the previous methods are struggling for "over-adapt to the

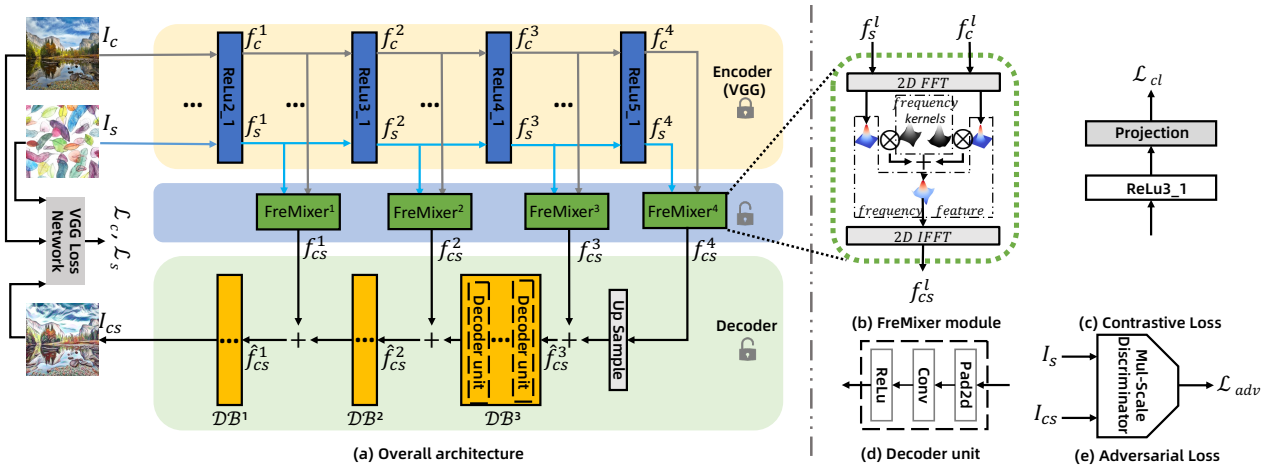

Figure 2: An overview of the Frequency Domain Disentanglemen (FDD) framework. (a) displays the overall architecture, which mainly contains a pre-trained encoder, a series of FreMixer modules, and a decoder. The content loss $\mathcal{L}_c$ and the style loss $\mathcal{L}_s$ force the model to learn the content information and reference style pattern. (b) and (d) illustrate the detailed structure of the proposed FreMixer module and the decoder unit, respectively. The decoder block ($\mathcal{DB}$) is a cascade of several decoder units. (c) and (e) depict the contrastive loss and adversarial loss that are used to learn the style-to-style relations and render a more realistic stylized image.

reference domain" and distorting the structural information. To force the model to maintain the structure of the object, the recent work Frequency Domain Image Translation (FDIT) computes the reconstruction loss regulating the frequency consistency to preserve the identity in the frequency domain. The FSP-based methods achieve significant improvements and inspire us to conduct arbitrary neural style transfer from the frequency perspective.

Methodology

We first describe the overall framework, then introduce the details of FreMixer module and the loss function for training the network.

Overview of the FDD Framework

As shown in Figure 2, the content image I_c and the style image I_s are fed into the proposed network to render the stylized image I_{cs}. We adopt an encoder-decoder style architecture network, which contains a frozen encoder Enc, a series of FreMixer modules and a decoder Dec. We employ the pre-trained VGG-19 (Simonyan and Zisserman 2014) as the encoder to extract features $f_c^{1:4}$ and $f_s^{1:4}$ on $\Psi^{1:4}$ (*ReLu2_1*, *ReLu3_1*,*ReLu4_1*,*ReLu5_1*) layers, respectively:

$$f_*^{1:4} = Enc(I_*) = \Psi^{1:4}(I_*), \tag{1}$$

where $*$ represents the symbol c or s. The l^{th} FreMixer module $\mathcal{FM}^l$ produces the stylized feature by disentangling and re-entangling the content and style in the frequency domain:

$$f_{cs}^l = \mathcal{FM}^l(f_c^l, f_s^l), \tag{2}$$

where $l \in (1, 2, 3, 4)$. Following IEST (Chen et al. 2021), f_{cs}^3 and f_{cs}^4 are jointly fed into the 3^{rd} decoder block $\mathcal{DB}^3$ to fully exploit the high-level semantics:

$$\hat{f}_{cs}^3 = f_{cs}^3 + UpSample(f_{cs}^4), \tag{3}$$

where $UpSample$ denotes the nearest interpolation layer which up-samples the input feature f_{cs}^4 to the same shape of f_{cs}^3. In order to take full advantage of the multi-stage stylized features, f_{cs}^i and the output of decoder block $\mathcal{DB}^{i+1}$ are jointly fed into $\mathcal{DB}^i$:

$$\hat{f}_{cs}^i = f_{cs}^i + \mathcal{DB}^{i+1}(\hat{f}_{cs}^{i+1}), \tag{4}$$

where $i \in (1, 2)$. With multi-stage stylized features, we can render the stylized image I_{cs} with the decoder Dec as:

$$I_{cs} = Dec(\hat{f}_{cs}^1, \hat{f}_{cs}^2, \hat{f}_{cs}^3). \tag{5}$$

Fast Fourier Transform (FFT)

To make the paper self-contained, we briefly describe the basic concepts here. Firstly, we present the definition and conjugate symmetric property of discrete Fourier transform (DFT), which is particularly critical for the proposed FreMixer module. Let $x[n]$, $n \in [0, N-1]$ be the original signal with the length of N. The 1-D DFT is formulated as:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \tag{6}$$

where $j = \sqrt{-1}$ is the imaginary unit. As is well known, DFT has the conjugate symmetric property, i.e., $X[N-k] = X^*[k]$. Thus, the half of $X[k]$, $k \in [0, \lceil N/2 \rceil]$ is sufficient to recover the original signal. The complexity of evaluating 1-D DFT naively is $O(n^2)$ and FFT (Cooley and Tukey 1965) reduces the complexity to $O(nlogn)$.

Because 2-D FFT is often executed by performing 1-D FFT in the horizontal direction and vertical direction, 1-D

FFT can be extended to 2D signals like a gray image. Let $x[m,n]$, $m \in [0, M-1]$, $n \in [0, N-1]$ be a 2D signal, it can be converted to frequency domain by 2-D DFT:

$$X[u,v] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m,n] e^{-j2\pi(\frac{um}{M} + \frac{vn}{N})}. \quad (7)$$

2D DFT also maintain the conjugate symmetric property $X[M-u, N-v] = X^*[u,v]$. The description of the inverse FFT (IFFT) is skipped for the sake of brevity.

Frequency Mixer Module

The proposed FreMixer module generates the stylized feature by disentangling and re-entangling the content and style in the frequency domain.

Frequency Transfer: We perform 2-D FFT along the spatial dimensions to convert the spatial feature maps into frequency domain:

$$F_* = \mathcal{F}(f_*), \quad (8)$$

where $*$ represents the symbol c or s, $\mathcal{F}(\cdot)$ represents the 2-D FFT operation, and $F_* \in \mathbb{C}^{C \times H \times W}$ stands for the frequency spectrum of f_*.

Disentangling: We introduce learned frequency kernels $K_* \in \mathbb{C}^{C \times H \times W}$ which play the role of a global depth-wise convolution layer to disentangle the content and style frequency patterns from F_*:

$$\widetilde{F}_* = F_* \odot K_*, \quad (9)$$

where $\odot$ means Hadamard product, $\widetilde{F}_*$ are the disentangled frequency spectrums of content and style from F_c and F_s. $\widetilde{F}_c$ preserves the content information of I_c, and $\widetilde{F}_s$ contains the style information of I_s. Figure 6 shows different characteristics of K_c and K_s, and Figure 7 visualizes the corresponding image of $\widetilde{F}_c$ and $\widetilde{F}_s$.

Re-entangling: After disentangling the content and style frequency pattern, these two frequency spectrums are recombined by element-wise addition operation:

$$F_{cs} = \widetilde{F}_c + \widetilde{F}_s. \quad (10)$$

Inverse Frequency Transfer: The frequency spectrum F_{cs} is transferred to the stylized feature f_{cs} in the spatial domain by 2-D IFFT:

$$f_{cs} = \mathcal{F}^{-1}(F_{cs}), \quad (11)$$

where $\mathcal{F}^{-1}(\cdot)$ is the 2-D IFFT operation. Pytorch (Paszke et al. 2019) implementation of the FreMixer module is included in the supplementary material.

Because the feature map $f_* \in \mathbb{R}^{C \times H \times W}$ is a real signal, the complex tensor F_* is conjugate symmetric. Thus, we can reduce the computation cost by using the half of F_* to preserve the full information. When taking only the half of F_*, the parameters of K_* are reduced from $C \times H \times W$ to $C \times H \times \lceil W/2 \rceil$.

Loss Function

As shown in Figure 2, the overall loss function is the weighted summation of content loss $\mathcal{L}_c$, style loss $\mathcal{L}_s$, contrastive loss $\mathcal{L}_{cl}$ and adversarial loss $\mathcal{L}_{adv}$:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_c + \lambda_2 \mathcal{L}_s + \lambda_3 \mathcal{L}_{cl} + \lambda_4 \mathcal{L}_{adv}, \quad (12)$$

where the weighted hyper-parameters $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ are set to $\{2, 1, 0.15, 5\}$ in this study.

Content Loss: Similar to (Mahendran and Vedaldi 2015), Euclidean distance between the VGG-19 features of the content image I_c and stylized image I_{cs} is used to reconstruct the content of I_c:

$$\mathcal{L}_c = \sum_{i=2}^{3} \|\Psi^i(\mathcal{I}_{cs}) - \Psi^i(\mathcal{I}_c)\|_2, \quad (13)$$

where Ψ^i represents the i-th layer of the pre-trained VGG-19 network.

Style Loss: We also employ the style loss $\mathcal{L}_s$ to match the mean μ and standard deviations σ of VGG-19 features of the style image I_s and stylized image I_{cs}, in the same way as used in AdaIN (Huang and Belongie 2017):

$$\begin{aligned} \mathcal{L}_s = &\sum_{i=0}^{4} \|\mu(\Psi^i(\mathcal{I}_{cs})) - \mu(\Psi^i(\mathcal{I}_s))\|_2 \\ &+ \|\sigma(\Psi^i(\mathcal{I}_{cs})) - \sigma(\Psi^i(\mathcal{I}_s))\|_2. \end{aligned} \quad (14)$$

Style Contrastive Loss: We introduce the contrastive loss to learn the relation between the stylized images. It pulls the adjacent stylized images closer, in which the image shares the same style with each other, and pushes away the stylized images with different styles:

$$\mathcal{L}_{cl} = -log(\frac{exp(P(I_{cs})^T P(I_{cs}^+)/\tau)}{exp(P(I_{cs})^T P(I_{cs}^+)/\tau) + \sum exp(P(I_{cs})^T P(I_{cs}^-)/\tau)}), \quad (15)$$

following (Chen et al. 2021), P is the projection networks, $\tau = 0.2$ is the temperature parameter controlling push and pull force. Two styled images I_{cs} and I_{cs}^+ are the positive pair with the same style but different contents. Similarly, I_{cs} and I_{cs}^- are the negative pairs with different styles in the mini-batch. More details can be referred to (Chen et al. 2021).

Adversarial Loss: Generative Adversarial Networks (GAN) is a popular generative model, and we use it to learn more "realistic" style patterns from the style image set. As shown in Figure 2, we introduce the multi-scale discriminator $\mathcal{D}_i, i \in (1, 3)$ (Park et al. 2019) to distinguish True/False between I_s and I_{cs}:

$$\mathcal{L}_{adv} = \sum_{i=1}^{3} \mathbb{E}[log(\mathcal{D}_i(I_s))] + \mathbb{E}[log(1 - \mathcal{D}_i(I_{cs}))] \quad (16)$$

Experiments

In this section we conduct extensive experiments to evaluate the effectiveness of the proposed method. Firstly, we introduce the implementing details. Secondly, we make qualitative and quantitative comparisons with other SOTA methods. Thirdly, ablation studies with different settings are conducted to explore the effects of the components in our model.

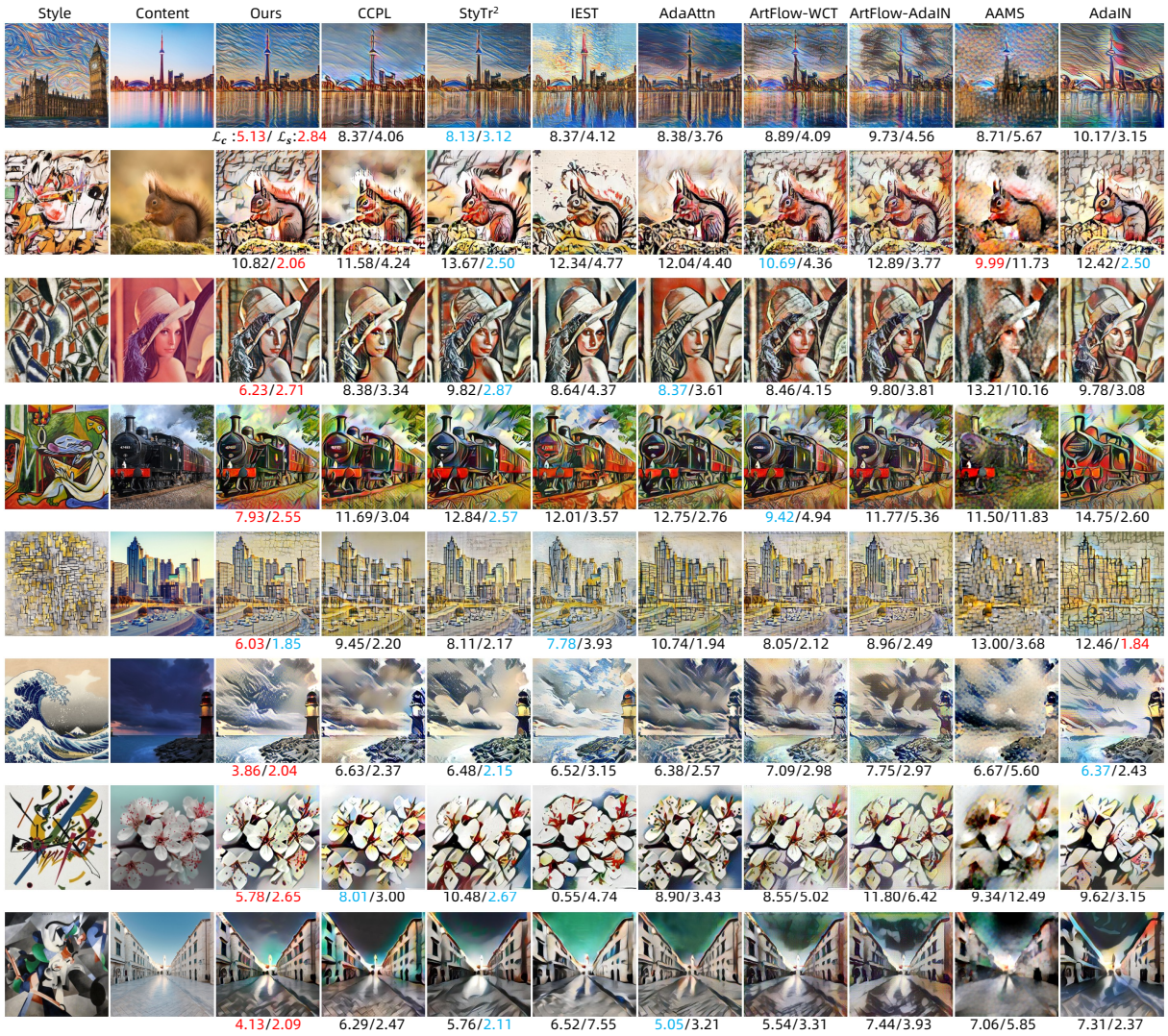

Figure 3: Comparison with other SOTA methods in arbitrary image style transfer. The first two columns show the style and content images. The rest of the columns show the stylized images of different methods. $\mathcal{L}_c$ and $\mathcal{L}_s$ are calculated for each stylized sample, with red and blue texts marking the best and second best, respectively.

Implementing Details

We use MS-COCO (Lin et al. 2014) as the content image set and use WikiArt (Phillips and Mackintosh 2011) as the style image set. During the training stage, all input images are first resized to 512×512 and then randomly cropped regions of size 256×256. Adam (Kingma and Ba 2014) with the learning rate of 0.0001 is used as the optimizer. We set the batch size to be 8 and train the proposed model with 160K iterations.

Comparison with SOTA Methods

We choose several SOTA methods as the benchmarks, including CCPL (Wu et al. 2022), StyTr^2 (Deng et al. 2022), IEST (Chen et al. 2021), AdaAttN (Liu et al. 2021), ArtFlow (An et al. 2021), AAMS (Yao et al. 2019) and AdaIN (Huang and Belongie 2017).

Qualitative Comparison: In Figure 3, we compare the stylized results of the proposed method and SOTA methods. It should note that all test images were excluded from the training data and all SOTA methods are reproduced by the officially open-sourced codes and pre-trained models.

The results show that only considering the global mean and variance alignment at the features level results in significantly corrupted content in AdaIN. Because AAMS only focuses on the main edge region, the most detail texture are lost. Although ArtFlow achieves better stylized results, the repetitive patterns (3^{rd} and 8^{th} rows) and loss of content details (2^{nd} and 4^{th} rows) are still frequent. Compared to the latest SOTA methods such as CCPL, StyTr^2, IEST, and AdaAttN, our method can preserve more structure information ($5th$ and $8th$ rows) and content details ($6th$ and $7th$ rows) with the similar stylization degree. In overview,

Method	Ours	CCPL	StyTr^2	IEST	AdaAttN	ArtFlow-WCT	ArtFlow-AdaIN	AAMS	AdaIN
SSIM↑	**0.51**	0.43	0.46	0.37	0.46	0.44	0.36	0.32	0.25
$\mathcal{L}_c \downarrow$	**7.91**	9.15	10.37	9.36	10.60	8.80	10.51	11.04	11.03
$\mathcal{L}_s \downarrow$	**2.40**	3.49	2.53	4.22	3.11	4.42	4.53	7.31	2.94
T_{256}(ms/img)↓	10.7	**7.4**	23.7	10.2	18.1	78.6	77.9	789.7	**7.0**
T_{512}(ms/img)↓	35.1	**23.7**	136.8	35.1	58.5	206.7	206.0	817.2	**23.1**
User study↑	**1102**	251	191	143	113	92	43	39	26

Table 1: Quantitative comparisons between the proposed method and other SOTA methods in terms of SSIM, $\mathcal{L}_c$ and $\mathcal{L}_s$, inference time, and user study. T_{256} represents the average inference time of 256 × 256 images.

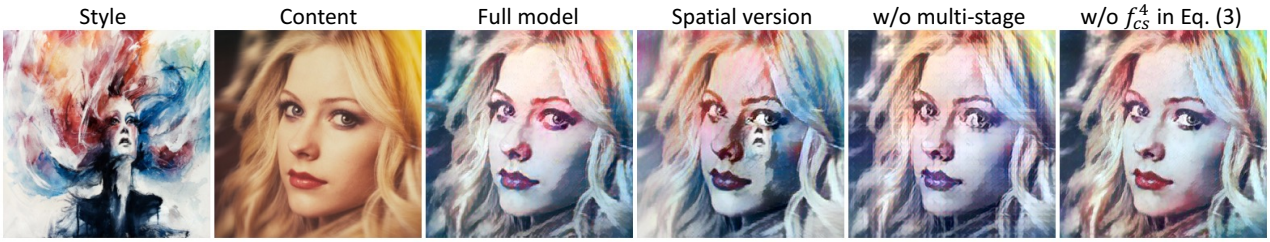

Figure 4: Ablation study on spatial-domain disentanglement(4^{th} col.), multi-stage strategy (5^{th} col.), and the highest-level semantics (6^{th} col.). Zoom-in for better view.

Method	SSIM↑	$\mathcal{L}_c \downarrow$	$\mathcal{L}_s \downarrow$
Full model	0.51	7.91	2.40
Spatial version	0.39	11.51	2.33
w/o multi-stage	0.44	8.93	3.20
w/o f_{cs}^4 in Eq. (3)	0.50	8.35	2.52

Table 2: Quantitative results for ablation study.

benefiting from clearer content and style disentanglement in the frequency domain, our proposed method can render the photo-realistic stylized image while preserving content structure information very well.

Quantitative Comparison: We further conduct quantitative comparison in terms of the SSIM, $\mathcal{L}_c$, $\mathcal{L}_s$, and inference time to demonstrate the superiority of the proposed method. For the first three metrics, we compute the average values of 2000 random stylized samples. For the inference time, all the models are tested on a single NVIDIA Tesla V100-32G with batch size 1. We run the model 2000 times and average the inference time.

-*SSIM*: SSIM is widely used to measure the structural similarity between two images. Following ArtFlow (An et al. 2021), we use it to measure the performance of the structural information preservation. As shown in Table 1, we achieve the best SSIM score, which demonstrates that the proposed method can preserve more structural information.

-$\mathcal{L}_c$, $\mathcal{L}_s$: Following StyTr^2 (Deng et al. 2022), we calculate $\mathcal{L}_c$ and $\mathcal{L}_s$ based on Equation (13) and Equation (14). Intuitively, the lower the value, the better the source content and style are preserved. The Table 1 shows that the proposed method achieves both the best scores.

-*Inference time*: As shown in Table 1, our method achieves faster or similar inference speed compared with the latest SOTA benchmarks such as StyTr^2, IEST, and AdaAttN. Although CCPL and AdaIN are a little faster than our method, the quality of their styled images is sacrificed. Our methhod achieves a better trade-off between the speed and the stylized image quality.

-*User study*: We invite 10 participants and randomly choose 200 samples for each participant. Each participant will be asked to choose his/her favorite result for each sample. We collect 2000 votes and show the votes in Table 1.

Ablation Study

As shown in Figure 4 and Table 2, we present ablation studies on *spatial-domain disentanglement*, *multi-stage strategy*, and the *highest-level semantics*. *Full model* contains all modules introduced in the section of **Methodology**. *Spatial version* replaces the FreMixer module with a standard 3×3 convolutional layer to disentangle the content and style, which is widely used in previous works. The result of *Spatial version* (the 4^{th} col in Figure 4) shows that the content of the style image is mixed into the stylized image. Additionally, since the stylized results contain part of the style image, Table 2 shows SSIM and $\mathcal{L}_c$ of *Spatial version* are the worst despite of the slightly lower $\mathcal{L}_s$. It demonstrates that its disentanglement of content and style in spatial domain is sub-optimal which results in the structural distortion in the stylized results. In *w/o multi-stage*, only f_{cs}^3 and f_{cs}^4

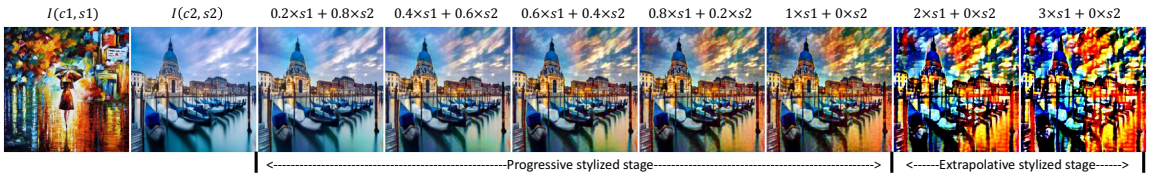

Figure 5: Progressive and extrapolative stylization. $I(c1, s1)$ is the combination of content $c1$ and style $s1$.

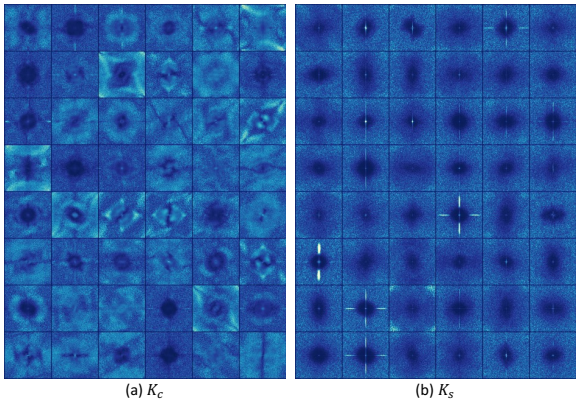

Figure 6: Visualization of the learnable frequency-domain kernels K_c (a) and K_s (b). All present kernels are learned by the proposed method. Zoom-in for better view.

are fed into the Decoder. We can observe that texture details are significantly lost despite the similar stylization is achieved. It concludes that shallow frequency features are essential to keep more detail information of the source content image while rendering photo-realistic stylized images. When removing the highest-level frequency feature in *w/o* f^4_{cs} *in Eq.(3)*, $\mathcal{L}_c$ and $\mathcal{L}_s$ are both slightly worse than the ones of the *Full model* in Table 2. Therefore, the highest-level frequency feature can also improve the quality of the stylized results.

Progressive and Extrapolative Stylization

We explore progressive and extrapolative stylization to achieve varying degrees of stylization by adjusting the style weights α and β of two images:

$$\widetilde{F}_s = \alpha s_1 + \beta s_2 = \alpha F_s \odot K_s + \beta F_c \odot K_s, \qquad (17)$$

where $\widetilde{F}_s$ is a combination of style frequency spectrum of content image and style image. As shown in Figure 5, when $\alpha + \beta = 1$, the source content image is progressively stylized, when $\beta > 1$, the source content image is over-stylized. The results demonstrate that the proposed method successfully disentangle content and style into independent components to control stylization.

Visualization of Kernels K_c and K_s

To analyze the frequency-domain characteristics of content and style, we visualize the learnable frequency-domain kernels K_c and K_s of the FreMixer module in Figure 6. We can observe that K_c and K_s have significantly different characteristics. In specific, K_c is similar to a band-pass or high-pass filter, while K_s is distributed from the low frequency band to the high frequency band. It demonstrates that our method can learn the different frequency pattern for content and style in the frequency domain.

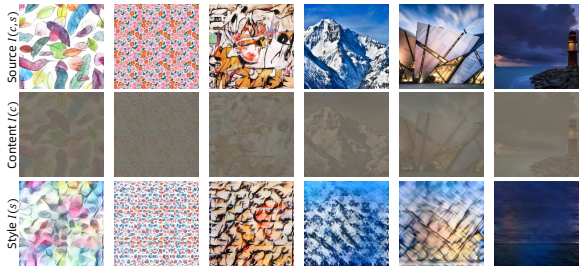

Figure 7: Visualization of the content and style components.

Visualization of the Content and Style Components

Compared with ArtFlow (An et al. 2021), we not only visualize the content $I(c)$ but also visualize style $I(s)$ in the image space to demonstrate the effective disentanglement in frequency domain in Figure 7. For a source image $I(c, s)$, we reconstruct its content $I(c)$ and style $I(s)$ by setting the frequency spectrum of style or content to 0, respectively. $I(c)$ preserves the original structure of the source image while removing the most style character. On the contrary, $I(s)$ contains the distinct style character (*e.g.,* colors and textures) while effacing the structure information. Figure 7 shows that the proposed method can clearly disentangle content and style in the frequency domain.

Conclusion

We propose a novel FreMixer module, which consists of the learned frequency kernels to disentangle and re-entangle the content and style components from a frequency perspective. As the different frequency-domain characteristics of the content and style components (*e.g.,* frequency bands and frequency patterns, see Figure 6) in the frequency domain, the proposed method could disentangle these two components more clearly and render the higher-quality stylized results (see Figure 3). Benefiting from the computationally efficient mechanism of the FreMixer, the speed of our FDD framework is comparable to SOTA approaches (see Table 1). The extensive experiments demonstrates the effectiveness of the proposed method and we believe our work will provide new insights to the community.

References

An, J.; Huang, S.; Song, Y.; Dou, D.; Liu, W.; and Luo, J. 2021. Artflow: Unbiased image style transfer via reversible neural flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 862–871.

Cai, M.; Zhang, H.; Huang, H.; Geng, Q.; Li, Y.; and Huang, G. 2021. Frequency domain image translation: More photo-realistic, better identity-preserving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13930–13940.

Chen, H.; Wang, Z.; Zhang, H.; Zuo, Z.; Li, A.; Xing, W.; Lu, D.; et al. 2021. Artistic style transfer with internal-external learning and contrastive learning. *Advances in Neural Information Processing Systems*, 34: 26561–26573.

Chen, Y.; Fan, H.; Xu, B.; Yan, Z.; Kalantidis, Y.; Rohrbach, M.; Yan, S.; and Feng, J. 2019. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3435–3444.

Cooley, J. W.; and Tukey, J. W. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, 19(90): 297–301.

Deng, Y.; Tang, F.; Dong, W.; Ma, C.; Pan, X.; Wang, L.; and Xu, C. 2022. StyTr2: Image Style Transfer with Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11326–11336.

Drori, I.; Cohen-Or, D.; and Yeshurun, H. 2003. Example-based style synthesis. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, II–143. IEEE.

Efros, A. A.; and Freeman, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 341–346.

Elad, M.; and Milanfar, P. 2017. Style transfer via texture synthesis. *IEEE Transactions on Image Processing*, 26(5): 2338–2351.

Frigo, O.; Sabater, N.; Delon, J.; and Hellier, P. 2016. Split and match: Example-based adaptive patch sampling for unsupervised style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 553–561.

Fu, X.; Huang, J.; Zeng, D.; Huang, Y.; Ding, X.; and Paisley, J. 2017. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3855–3863.

Fu, Y.-H.; Kang, L.-W.; Lin, C.-W.; and Hsu, C.-T. 2011. Single-frame-based rain removal via image decomposition. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1453–1456. IEEE.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2414–2423.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, 1501–1510.

Jing, Y.; Yang, Y.; Feng, Z.; Ye, J.; Yu, Y.; and Song, M. 2019. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11): 3365–3385.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, 694–711. Springer.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kotovenko, D.; Sanakoyeu, A.; Lang, S.; and Ommer, B. 2019. Content and style disentanglement for artistic style transfer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4422–4431.

Kyprianidis, J. E.; Collomosse, J.; Wang, T.; and Isenberg, T. 2012. State of the" art": A taxonomy of artistic stylization techniques for images and video. *IEEE transactions on visualization and computer graphics*, 19(5): 866–885.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Liu, S.; Lin, T.; He, D.; Li, F.; Wang, M.; Li, X.; Sun, Z.; Li, Q.; and Ding, E. 2021. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6649–6658.

Mahendran, A.; and Vedaldi, A. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5188–5196.

Park, D. Y.; and Lee, K. H. 2019. Arbitrary style transfer with style-attentional networks. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5880–5888.

Park, T.; Liu, M.-Y.; Wang, T.-C.; and Zhu, J.-Y. 2019. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2337–2346.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Perona, P.; and Malik, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7): 629–639.

Phillips, F.; and Mackintosh, B. 2011. Wiki Art Gallery, Inc.: A case for critical thinking. *Issues in Accounting Education*, 26(3): 593–608.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Wu, X.; Hu, Z.; Sheng, L.; and Xu, D. 2021. Styleformer: Real-time arbitrary style transfer via parametric style composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14618–14627.

Wu, Z.; Zhu, Z.; Du, J.; and Bai, X. 2022. CCPL: Contrastive Coherence Preserving Loss for Versatile Style Transfer. In *ECCV*.

Yang, Y.; and Soatto, S. 2020. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4085–4095.

Yao, Y.; Ren, J.; Xie, X.; Liu, W.; Liu, Y.-J.; and Wang, J. 2019. Attention-aware multi-stroke style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1467–1475.