A Comparative Study of CNN and Transformer based Visual Style Transfer

Huapeng Wei
School of AI, Jilin University
Changchun 130012, China
weihp20@mails.jlu.edu.cn

Yingying Deng
NLPR, CASIA
Beijing 100190, China
dengyingying2017@ia.ac.cn

Fan Tang*
School of AI, Jilin University
Changchun 130012, China
tangfan@jlu.edu.cn

Xingjia Pan
Youtu Laboratory, Tencent
Shanghai 200233, China
noahpan@tencent.com

Weiming Dong
NLPR, CASIA
Beijing 100190, China
weiming.dong@ia.ac.cn

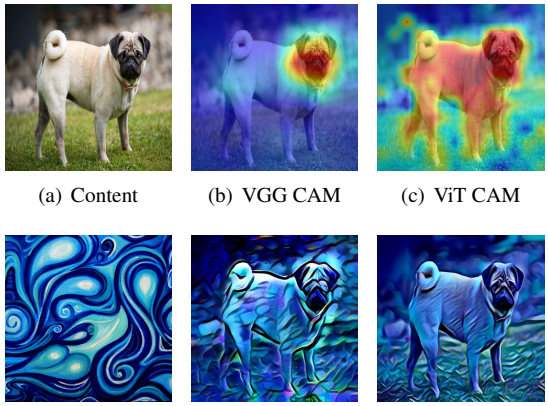
Abstract

Vision Transformers have shown impressive performance on the image classification task. Observing that most existing neural style transfer algorithms are based on texture-biased CNNs pre-trained on the image recognition task with ImageNet dataset, here raises the question that whether the shape-biased vision Transformers can perform style transfer as CNNs. In this work, we focus on comparing and analyzing the shape bias between CNNs and Transformer with our proposed Transformer-based visual style transfer methods (Tr-NST, Tr-AdaIN, Tr-WCT). We show that Transformers pre-trained on ImageNet are not proper for typical style transfer methods due to the strong shape bias from both learned parameters and the structure design. By comparing Transformers and CNNs on the view of visual style transfer via experiments variations, we provide evidence that when retrained with proper style supervision, Transformers can learn similar features as CNNs which capture local textures and style patterns, alleviate the shape bias from the learned parameters. Qualitative experiments demonstrate that the proposed Tr-AdaIN can generate comparable results with the state-of-the-art visual style transfer methods.

***Keywords:** transformer, convolution neural network, visual style transfer, comparative study*

1. Introduction

Visual style transfer (VST) refers to the task which renders given visual media into the desired style while preserving its content structure. Since Gatys *et al*. [11] proposed the Gram matrix based style measurement, researchers have been focusing on performing style transfer with convolutional neural networks (CNNs) in an end-to-end manner. Optimization-based or feed-forward approaches are two kinds of mainstream VST approaches focusing on directly stylizing input visual media without involving specific domain knowledge of arts. Optimization-based methods [11, 17] achieve style transfer in an iterative optimization process, which produce high-quality images but are computationally expensive due to the direct backward on image pixels. Feed-forward approaches [14, 21, 8, 7] focus on arbitrary or universal style transfer using encoder-decoder pipeline to generate visual media with remarkable quality for different styles. All of these methods are based on features extracted from CNNs which are pre-trained on image classification tasks.



(a) Content (b) VGG CAM (c) ViT CAM

(d) Style (e) AdaIN (f) Tr-AdaIN

Figure 1. Visual comparisons for CNN- and Transformer-based structures. From the class activation map (CAM, (b) and (c)), we could observe that the pre-trained ViT prefers to make prediction based on integral shape, whereas the VGG focuses on local patch, showing that ViTs have stronger shape bias than CNNs. The bottom row shows the style transfer results generated by AdaIN (e) and the proposed Tr-AdaIN (f).

*Corresponding author.

Recently, Transformer [32] provides an alternative architecture for CNNs in the research field of visual media computing. Researchers adopted the sequence to sequence architecture to the field of visual media computing. Represented by Vision Transformer (ViT) [9], Transformer-based models outperform CNNs in various CV tasks [3, 38, 19, 13]. The success of Transformer has heightened the need for clearing the properties of Transformer. Overall, researchers consider the difference of ViT and CNN backbone caused by the long-range dependencies and local-receptive field. In-depth comparisons point out that CNNs are texture biased while Transformers are shape biased, as shown in Figs. 1(b) and 1(c).

Tuli *et al*. [30] compared CNNs and Transformers on the view of human vision similarity. By measuring and analyzing the error consistency for different networks, they found that in contrast to CNNs, Transformers behave higher shape bias and are largely closer with human vision. Their experiments showed that Transformers can maintain the accuracy and increase the shape bias when fine-tuned on augmented data, whereas CNNs drop performance at the same time. Naseer *et al*. [24] studied the shape and texture bias of Transformers and CNNs. They also pointed out that the strong shape bias leads the Transformers to perform better than CNNs and comparable to humans on shape recognition. Besides, they found that the shape bias makes the Transformers more robust towards domain shifts.

However, all these conclusions are made upon image classification tasks. Few studies have investigated comparisons between CNN and Transformers on generative tasks, such as style transfer. Shape and texture, corresponding to content and style, are considered to be the basic constituent elements in the VST process. As shown in Figs. 1(e) and 1(f), when using similar configurations for the transfer module, different types of backbones lead to different VST results.

To figure out the cause of shape bias in Transformer, we attempt to engage the architecture into typical VST algorithms (NST, AdaIN, and WCT), visiting the following fundamental problem which is central both to VST and the research field of computational visual media: is the shape bias of Transformer from the model parameters or structure. Researchers widely hold the point that different structures lead to different results. Comparing the results of Transformer-based VST, we found that the typical VST approaches do not work well on Transformer structure. However, by further controlling the training configuration of Transformer encoder, we prove that the same structure may generate totally different results when the model parameters are different. Overall, we conduct preliminary comparisons among CNN- and Transformer-based VST approaches which may through new light upon the research field of deep structure based approaches. Our contributions are summarized as follows:

- We engage three typical VST approaches with Transformer by proposing Tr-NST, Tr-AdaIN, and Tr-WCT. Results show that pre-trained ViTs are invalid for performing mainstream style transfer methods due to the shape bias.
- We demonstrate that the shape bias can be reduced by training with proper supervision. ViTs can obtain the ability to capture style strokes and patterns, which leads the models towards texture bias instead of shape bias.
- We discuss the influence of basic Transformer modules such as position encoding, and upsampling way for Transformer based VST tasks.

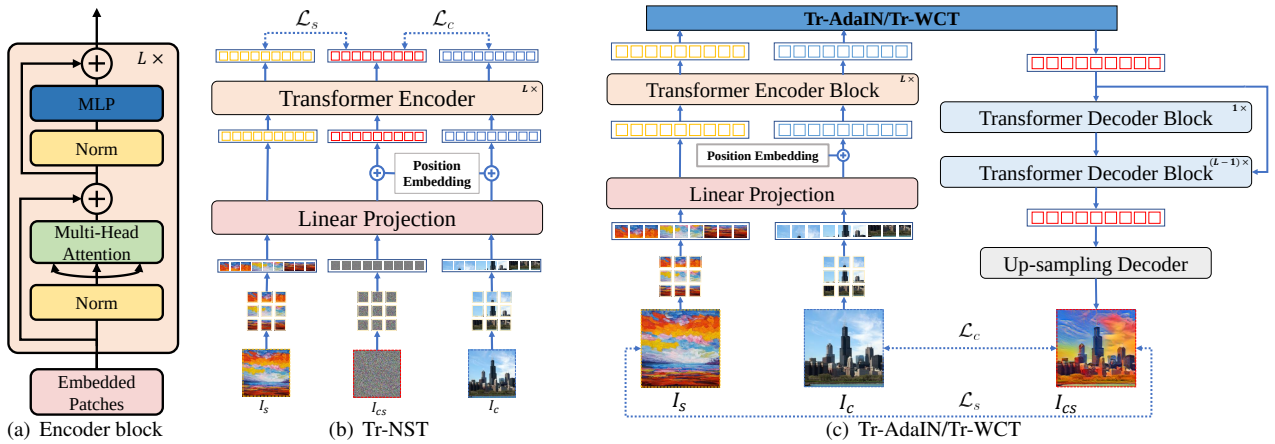
2. Related works

2.1. Neural Style transfer

Style transfer is a long-standing task in the image process field. Gatys *et al*. [11] open up the neural style transfer area, introducing the style representation by computing Gram matrix upon deep features extracted by convolution neural networks. Johnson *et al*. [16] and Ulyanov *et al*. [31] train a single feed-forward network to transfer specific style to arbitrary content, achieve faster inference speed than directly optimizing pixels. But these models need to be trained from scratch to render a new style. Li *et al*. [21] propose the whiting and coloring transformation, namely WCT, for arbitrary style transfer. Huang *et al*. [14] introduce the adaptive instance normalization, namely AdaIN, which performs normalization on content feature with mean-variance statistics from style feature. Most recently, An *et al*. [1] propose ArtFlow based on reversible neural flow, alleviating the content leak phenomenon. Although recent developments [25, 7, 20, 34, 8] in the field of style transfer have made great achievements towards arbitrary style and high quality, these style transfer models are still almost based on deep features from CNNs. In this paper, we propose to perform style transfer with Vision Transformers.

2.2. Visual Transformers

Transformer [32] is proposed for machine translation task and has achieved the SoTA performance in various NLP tasks. Utilizing attention mechanisms, Transformer-based models can process long-range dependencies, aggregate information from all the tokens. Inspired by the success of Transformer-based models in the NLP field, there have been many attempts on adapting the Transformer architecture to computer vision tasks. Chen *et al*. [5] introduce a sequence Transformer pre-trained to auto-regressively predict pixels. Proposed by Dosovitskiy *et al*. [9], Vision Transformer (ViT) has achieved SoTA performance on image



(a) Encoder block (b) Tr-NST (c) Tr-AdaIN/Tr-WCT

Figure 2. Model architectures for the three proposed Transformer based VST approaches.

classification tasks, outperforms previous CNNs. With a pure Transformer architecture, ViT process images patches as tokens directly, instead of pixels. In addition to the image classification task, Transformers have been used to solve other various computer vision tasks [37], such as object detection [3], segmentation [33], image processing [4, 18, 23], image generation [15, 19] and 3D computer vision [13]. In this paper, we investigate the ability of Transformer on representing style and preserving the content structure, explore the property by performing style transfer with Transformer.

2.3. Comparative study of CNN and visual Transformers

Analyses of CNN have been proposed a lot due to its high performance in visual tasks. Following the breakthrough of Transformer in computer vision field, there exist works paying attention to horizontal comparison between CNNs and visual Transformers. Cordonnier *et al.* [6] prove that self-attention layers can express any convolutional layer when given sufficiently many heads. Besides, they give proof that fully-attentional models seem to learn a generalization of CNNs where the kernel pattern is learned at the same time as filters. Tuli *et al.* [30] explore that Transformers have higher shape bias than CNNs for image classification tasks, conclude that attention models focus on the important part of the image for the given task, and neglect the otherwise noisy background to make predictions. Naseer *et al.* [24] get similar conclusions with Tuli. They analyze the one properties of ViT that Transformers are more biased towards shape compared to CNNs, and this leads Transformers highly robust to severe occlusions, perturbations, and domain shifts for the image classification task. Both of the above works on comparison between Transformers and CNNs explore important properties for the image classification tasks without considering the effect on generative tasks. In this paper, we are mainly concerned with properties of Transformers for generative tasks by horizontal comparing and analyzing Transformers and CNNs.

3. Transformer based VST

3.1. Formulations

Given a content image $I_c \in \mathbb{R}^{H\times W\times 3}$ and a style image $I_s \in \mathbb{R}^{H\times W\times 3}$, our goal is to render style from I_s to I_c for generating I_{cs}. We split the input content image I_c and style image I_s into patches, and then feed these patches into a linear projection layer to obtain sequence tokens $\mathcal{E}_c$ and $\mathcal{E}_s$ with the shape of $L \times C$, where $L = \frac{H\times W}{m\times m}$ refers to the token numbers, m refers to patch size, C refers to token's feature dimension.

Before being fed into Transformer encoder blocks, the sequence tokens can be added with an optional positional embedding. We adopted sinusoidal position embedding (SPE) [32] and learnable position embedding (LPE) [9] as alternative position encoding methods. The SPE can be calculated as:

$$\begin{aligned} SPE(pos, 2i) &= \sin(\frac{pos}{10000^{2i/C}}), \\ SPE(pos, 2i+1) &= \cos(\frac{pos}{10000^{2i/C}}). \end{aligned} \tag{1}$$

The LPE has the same shape with sequence tokens $\mathcal{E}$, which can also be added to the tokens to retain positional information. Given a sequence tokens $\mathcal{E}$ added with optional position embedding, each encoder block can be denoted in Fig.2(a). The multi-head self-attention can be calculated as:

$$\begin{aligned} &MultiHeadAttention(Q, K, V) = \\ &Concat(head_1(Q, K, V), ..., head_h(Q, K, V))W^o \\ &\text{where } head_i = Attention(QW_i^q, KW_i^k, VW_i^v), \end{aligned} \tag{2}$$

where $i = 1, ..., h$, h refers to the number of attention heads, $W_i^q \in \mathbb{R}^{C\times C}$, $W_i^k \in \mathbb{R}^{C\times C}$, $W_i^v \in \mathbb{R}^{C\times C}$ and

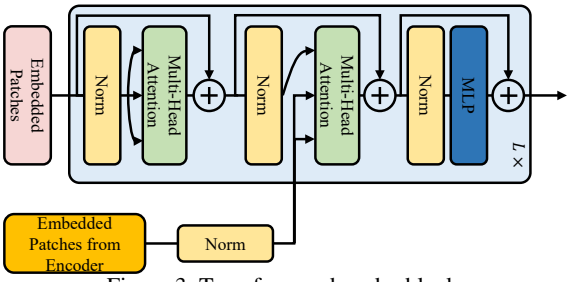


Figure 3. Transformer decoder block.

$W_i^o \in \mathbb{R}^{C\times C}$ are learnable parameter matrices for projections. The complete Transformer encoder block can be formulated as:

$$\begin{aligned} X &= MultiHeadAttention(\mathcal{E}_i, \mathcal{E}_i, \mathcal{E}_i) + \mathcal{E}_i, \\ X' &= FFN(X) + X, \end{aligned} \tag{3}$$

where FFN refers to the feed-forward network, $i = 1, ..., N_l$ is the layer index.

Similarly, each decoder block (as shown in Fig.3) can be formulated as:

$$\begin{aligned} Y &= MultiHeadAttention(\mathcal{E}_j, \mathcal{E}_j, \mathcal{E}_j) + \mathcal{E}_j, \\ Y' &= MultiHeadAttention(Y, \mathcal{Z}, \mathcal{Z}) + Y, \\ Y'' &= FFN(Y') + Y', \end{aligned} \tag{4}$$

where $\mathcal{Z}$ refers to the encoded sequence tokens from the Transformer encoder when reconstructing images or Tr-AdaIN and Tr-WCT when performing style transfer, $j = 1, ..., N_l$ is the layer index. We apply the Layer Normalization in the way of Pre-LN [36].

3.2. Tr-NST

We follow Gatys *et al.* [11] to perform optimization-based style transfer by proposing Tr-NST. As shown in Fig.2(b), instead of extracting features from a pre-trained VGG [28], we utilize the Transformer encoder features to compute perceptual loss. Generally, the feature sizes from different levels are the same for specific input. The content perceptual loss based on Transformer features is defined as:

$$\mathcal{L}_c = \frac{1}{N_l \times CL} \sum_{i=0}^{N_l} \|re(\phi_i(\mathcal{E}_{cs})) - re(\phi_i(\mathcal{E}_c))\|_2, \tag{5}$$

where $re(\cdot)$ is used to reshape the $\mathcal{E}$ into the shape of $C \times \frac{H}{m} \times \frac{W}{m}$ and $\phi_i(\cdot)$ denotes the features extracted from the i_{th} level Transformer encoder block. Similarly, we define the style perceptual loss based on Transformer feature as:

$$\mathcal{L}_s = \frac{1}{N_l \times CL} \sum_{i=0}^{N_l} \|G(\phi_i(I_{cs})) - G(\phi_i(I_c))\|_2. \tag{6}$$

where $G_i(\cdot) \in \mathbb{R}^{C\times C}$ denotes i_{th} level feature's Gram matrix, which is defined as:

$$G_i(\mathcal{E}) = \sum \mathcal{E}^{\mathrm{T}}\mathcal{E}. \tag{7}$$

The total perceptual loss based on Transformer feature is defined as:

$$\mathcal{L} = \mathcal{L}_c + \lambda\mathcal{L}_s. \tag{8}$$

We optimize an image $I_{cs} \in \mathbb{R}^{H\times W\times 3}$ using $\mathcal{L}$ where the $\mathcal{L}_c$ and $\mathcal{L}_s$ are computed with pairs (I_c, I_{cs}) and (I_s, I_{cs}), respectively. In our experiments, we use features extracted from the last five blocks in the ViT.

3.3. Tr-WCT

To perform arbitrary style transfer with Transformer, we add an additional Transformer decoder for the style and content representation coupling and reconstruction. The decoder shares the same data shape as the encoder. Detailed Transformer decoder block has been introduced in Sec.3.1.

Following the single-level WCT [21], we also perform the whitening and coloring transform in feature space. Given the content sequence embedding $\mathcal{E}_c$ and style embedding $\mathcal{E}_s$, we first center $\mathcal{E}_c$ and then obtain the whitened content feature $\hat{\mathcal{E}}_c$ by:

$$\hat{\mathcal{E}}_c = E_c D_c^{-\frac{1}{2}} E_c^{\mathrm{T}} \mathcal{E}_c, \tag{9}$$

where D_c is a diagonal matrix with the eigenvalues of the convariance matrix $\mathcal{E}_c\mathcal{E}_c^{\mathrm{T}} \in \mathbb{R}^{C\times C}$, and E_c is the corresponding orthogonal matrix of eigenvectors, satisfying $\mathcal{E}_c\mathcal{E}_c^{\mathrm{T}} = E_c D_c E_c^{\mathrm{T}}$. After centering $\mathcal{E}_s$, the colored feature $\hat{\mathcal{E}}_{cs}$ can be obtained by performing coloring transform, which is defined as:

$$\hat{\mathcal{E}}_{cs} = E_s D_s^{-\frac{1}{2}} E_s^{\mathrm{T}} \hat{\mathcal{E}}_c, \tag{10}$$

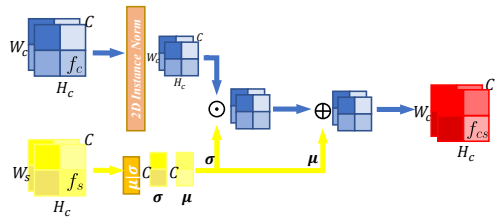
where D_s is a diagonal matrix with the eigenvalues of the convariance matrix $\mathcal{E}_s\mathcal{E}_s^{\mathrm{T}} \in \mathbb{R}^{C\times C}$, and E_s is the corresponding orthogonal matrix of eigenvectors. Finally, the $\hat{\mathcal{E}}_{cs}$ is re-centered by adding the mean vector m_s of the style:

$$\hat{\mathcal{E}}_{cs} = \hat{\mathcal{E}}_{cs} + m_s. \tag{11}$$

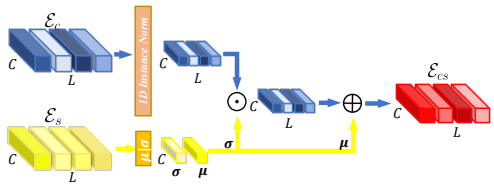
Then $\hat{\mathcal{E}}_{cs}$ is fed into the Transformer decoder to obtained stylized image. When trained with a pre-trained ViT for reconstructing an input image, we use the pixel reconstruction loss [10]:

$$\mathcal{L} = \|I_o - I_i\|_2^2, \tag{12}$$

where I_i, I_o are the input image and reconstruction output. When retraining the whole auto-encoder with the strategy introduced in Sec.4.3, we use the perceptual loss introduced in Sec.3.2 to alleviate the shape bias.



(a) AdaIN



(b) Tr-AdaIN

Figure 4. AdaIN and Tr-AdaIN. For AdaIN, C refers to the channel number of CNN features; for Tr-AdaIN, C refers to the feature dimension of tokens.

3.4. Tr-AdaIN

Another mainstream arbitrary style transfer method is adaptive instance normalization (AdaIN) [14]. We propose the Tr-AdaIN by replacing the Tr-WCT transform with AdaIN and retaining the decoder architecture. Given the content feature $\mathcal{E}_c$ and style feature $\mathcal{E}_s$, the output stylized feature $\mathcal{E}_{cs}$ from AdaIN is computed as $\mathcal{E}_{cs} = AdaIN(\mathcal{E}_c, \mathcal{E}_s) = \sigma_{\mathcal{E}_s}(\frac{\mathcal{E}_c - \mu_{\mathcal{E}_c}}{\sigma_{\mathcal{E}_c}}) + \mu_{\mathcal{E}_s}$, where $\sigma_{\mathcal{E}_c}$, $\mu_{\mathcal{E}_c}$, $\sigma_{\mathcal{E}_s}$ and $\mu_{\mathcal{E}_s}$ refer to dimension-wise variance and mean of $\mathcal{E}_c$ and $\mathcal{E}_s$, respectively. The detailed comparison against AdaIN and Tr-AdaIN is also illustrated in Fig. 4.

For the model optimization, we use an additional pre-trained VGG to support matching the mean and variance of VGG features. The content loss is defined as $\mathcal{L}_c = \frac{1}{N_l}\sum_{i=0}^{N_l} \|(\Phi_i(\mathcal{E}_{cs})) - (\Phi_i(\mathcal{E}_c))\|_2$, where $\Phi(\cdot)$ denotes output feature of the VGG relu layer.

The style loss is defined as:

$$\mathcal{L}_s = \frac{1}{N_l}\sum_{i=0}^{N_l} \|\mu(\Phi_i(I_{cs})) - \mu(\Phi_i(I_s))\|_2 + \sum_{i=0}^{N_l} \|\sigma(\Phi_i(I_{cs})) - \sigma(\Phi_i(I_s))\|_2, \tag{13}$$

where $\mu(\cdot)$ denotes the mean of features, and $\sigma(\cdot)$ denotes the variance of features.

$$\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_s. \tag{14}$$

In our experiments we use relu1_1, relu2_1, relu3_1, relu4_1 and relu5_1 layers of the pre-trained VGG-19.

4. Experiments

4.1. Setup

Datasets. We use MSCOCO [22] and WikiArt [26] as content and style dataset, correspondingly. During the training stage, each image is resized to 512×512 and then randomly cropped into 256×256 by default. The image can be any size during the test stage.

Implementation details. We use two NVIDIA RTX 3090 cards to train our model using the Adam optimizer with $lr = 5e^{-4}$. For Tr-AdaIN and Tr-WCT, the model is trained for 160000 iterations. The warmup strategy is used during the first 10000 iterations. For Tr-NST, the I_{cs} is trained for 4000 iterations. We use $\lambda = 1$ in Equation 8 and Equation 14 for perceptual loss; for the decoder training in Tr-WCT, we use a single pixel reconstruction loss. We counted the inference speed of each method, as shown in Table 1.

Table 1. Inference speed (FPS) and FLOPs of different methods on image size 512×512. For NST and Tr-NST, the FLOPs only counts single iteration; the inference speed describes how many iterations per second. For other methods, the inference describes frames per second (FPS) and the FLOPs are computed for single pair of content and style images.

Methods	Frames per second (FPS)	FLOPs
AdaIN	7.204	266.693G
Tr-AdaIN	5.666	231.628G
WCT	4.225	189.991G
Tr-WCT	3.281	231.628G
NST	10.269	63.339G
Tr-NST	9.922	87.578G

Different from traditional VST approaches where the encoder is pre-trained on image classification tasks and not involved in the training process, we discuss the differences of CNN-based and Transformer-based VST approaches by controlling the parameter training of the encoder. Given the Transformer-based visual style transfer methods described in Sec.3, we implement Tr-NST-p, Tr-AdaIN-p, and Tr-WCT-p by using pre-trained ViT as the encoder. The training process only influences the decoder for Tr-AdaIN-p and Tr-WCT-p. We further implement Tr-NST, Tr-AdaIN, and Tr-WCT by random initializing and training the encoder. The encoder of Tr-AdaIN and Tr-WCT is optimized based on the losses calculated on pre-trained VGG. After training, we use the encoder of Tr-AdaIN for Tr-NST. In general, the parameters of the encoder in Tr-VST and Tr-VST-p are affected by a pre-trained Transformer and a pre-trained CNN respectively. Since researchers have pointed out that the self-attention module can mimic a convolution layer [6], we try to figure out whether the influence of the Transformer's

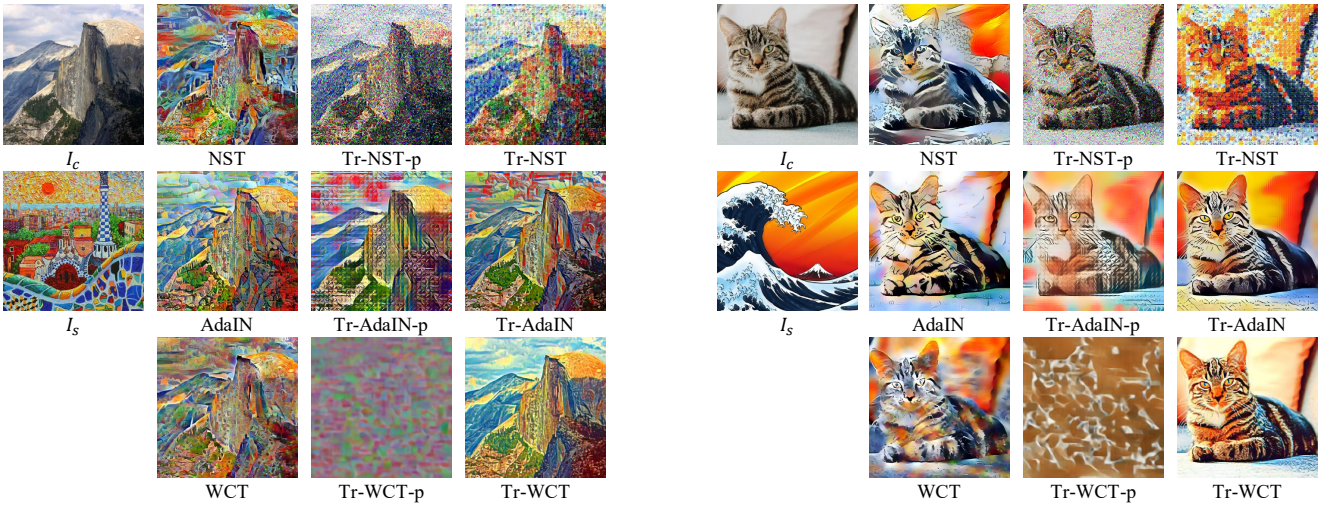


Figure 5. Visual style transfer results using different methods.

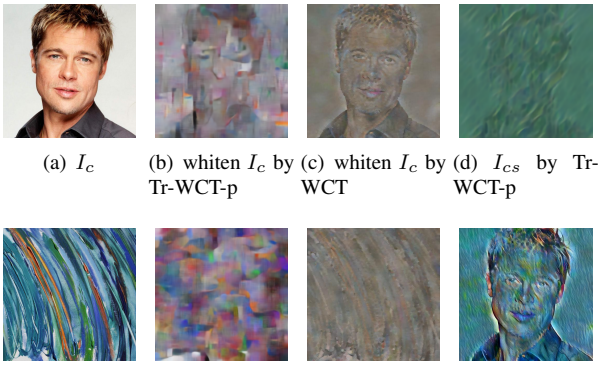
structure could be reduced with a properly set training configuration.

4.2. Are CNN-based VST approaches proper for Transformer?

In this section, we discuss the influence of adopting the Transformer structure for VST tasks. As shown in Fig.5, the CNN-based methods (NST, AdaIN, and WCT) can generate stylized images with high quality, but the methods adopted for Transformer (Tr-NST-p, Tr-AdaIN-p, and Tr-WCT-p) fail to some extent. Proposed by Gatys *et al*. [11], CNN-based NST is regarded as the benchmark in this field. However, the Tr-NST-p fails to render any style elements from the style image, and disturbs pixels and produces boundaries between patches. In the sense of style transfer, Tr-NST-p behaves no effect at all. AdaIN [14] provides a simple but effective way to match the style factors to the content image. Although Tr-AdaIN-p can also transfer the styles, the results appear boundary artifacts and unreasonable patterns.

WCT vs Tr-WCT-p. As a learning-free transformation, WCT can be performed with a reconstruction auto-encoder, generating high-quality stylized images in a coarse-to-fine manner. In this paper, we only consider single-level stylization. Compared to WCT, the Tr-WCT-p generates unexpected results, only retaining the incomplete outlines from the content image and little style pattern from the style image. Such results motivate us to go through the intermediate results of Tr-WCT-p.

To test and verify the conclusion, we reconstruct images with whitened features, following the original WCT. The whiten results generated with Tr-WCT-p and VGG are shown in Fig.6. When performed on VGG features, the whitening transform removes style information by removing the correlation between each channel. But this operation fails to disentangle style and content on Transformer features which do not provide sufficient style information. The stylized images from CNN can render the style from the reference image and save the content simultaneously, where the Transformer-based WCT can't even preserve the detailed content, only maintaining the blurred outline.



(a) I_c (b) whiten I_c by Tr-WCT-p (c) whiten I_c by WCT (d) I_{cs} by Tr-WCT-p

(e) I_s (f) whiten I_s by Tr-WCT-p (g) whiten I_s by WCT (h) I_{cs} by WCT

Figure 6. Stylized results and inverted features for WCT and Tr-WCT-p.

All the phenomena mentioned above show that Transformers pre-trained on ImageNet have strong shape preference. We argue that pre-trained ViT is not valid for extracting features to perform style transfer, due to the strong bias from both parameters learned from the classification task and the structure design. On the one hand, Transformers can obtain long-distance dependence, i.e. global receptive field with the help of multi-head attention mechanism. The

multi-head attention mechanism is position-agnostic, which computes the attention score over all the sequence tokens, focusing on the relationships between tokens. These relationships between image patches are often reflected in form of shape. Although Cordonnier *et al*. [6] proves that a self-attention can express any convolution, but the classification task doesn't provide valid a guide to that mimic, making the strong bias and the reconstruction error. On the other hand, all the pixels in the same block are treated as a token, losing both the spatial information and texture detail. In the next section, we will give a simple but efficient solution to alleviate this shape bias phenomenon.

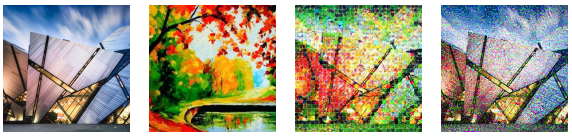
4.3. Is the shape bias from model parameters or model structure?

A shape-biased model prefers to make predictions relying more on shape; on the contrary, a texture-biased model makes predictions towards texture more than shape. Baker *et al*. [2] introduce that CNNs have been shown to have stronger texture bias rather than shape bias in the image classification task. Tuli *et al*. [30] introduce that ViT pre-trained on ImageNet dataset has a stronger shape bias than traditional CNNs (like ResNet). The ViT family and ResNet are trained on ImageNet-21K and ILSVRC-2012[27] datasets. When testing on the Stylized ImageNet[12], ResNet shows poorer performance whereas ViT can maintain part of accuracy. Naseer *et al*. [24] find that ViTs with higher shape bias not only perform better than ResNet but also more robustly. Following DeiT[29], they obtain a ViT with stronger shape bias with shape distillation training strategy.

For VST tasks, we tend to seek a texture-biased model, rather than a shape-biased one. A model with strong texture bias can capture various style elements like patterns and strokes, thereby representing style factors. For Transformer-based VST approaches, we suppose that part of the shape bias comes from the parameters, while the model structure provides an additional part on this trend. We propose Tr-NST, Tr-AdaIN, and Tr-WCT by adding constraints to the Transformer encoders by an additional pre-trained VGG.

We show the detailed qualitative examples in Fig.7. The Tr-AdaIN succeeds to fuse the content feature $\mathcal{E}_c$ with style feature $\mathcal{E}_s$ from the Transformer encoder. The decoder in Tr-AdaIN maps the fused feature $\mathcal{E}_{cs}$ back to the image domain, generating the stylized image I_{cs}. Compared to the failed cases generated by Tr-AdaIN-p, the Tr-AdaIN outperforms original AdaIN results based on pre-trained VGG. By performing Tr-WCT, we obtained the stylized images which also achieve comparable quality with original WCT based on pre-trained VGG.

Tr-NST vs Tr-NST-p. As shown in Fig. 5, results of Tr-NST are still not visual good compared with original NST. Fig. 8 shows another example. The Tr-NST-p fails to extract required style information, and the optimization only adds noise on the content image. In contrast, Tr-NST can optimize the content image using the Transformer encoder in Tr-AdaIN, rendering and coloring towards the stylized image. The results from both Tr-NST and Tr-NST-p appear boundary artifact between each image patch. This observation can be attributed to the reason that all pixels within a certain patch are optimized towards the same direction.



(a) I_c (b) I_s (c) Tr-NST (d) Tr-NST-p

Figure 8. Visual comparisons against Tr-NST and Tr-NST-p.

Comparison with other transformer-based VST works. To date, several previous studies have investigated performing VST with transformers [25, 35]. SANet [25] introduced attention mechanism as style transfer module, utilizing normalized content and style features to calculate the attention map. StyleFormer [35] improved the transformer-driven style composition approach which can learn content-consistent style composition for the style transfer task. SANet and StyleFormer achieve the SOTA stylization results, but both these works are still based on features extracted from a fixed VGG network. The Tr-VST methods are designed to investigate the discrepancy of representing style patterns between the transformers and CNNs. For our Tr-VST, the transformers are used as feature extractors instead of the style transfer module to be compared with VGG-based VST approaches.

4.4. Is position embedding fitting for style transfer with Transformer?

The multi-head self-attention layer is in a position-agnostic manner and can not naturally make use of the position in sequence tokens. Researchers introduce position embedding to provide relative or absolute positional information. Vaswani *et al*. [32] introduce the sinusoidal positional encoding to inject positional information. Dosovitskiy *et al*. [9] add a 1D learnable embedding to the tokens to retain the sequence order.

We test SPE and LPE which are introduced in Sec. 3.1 on Tr-AdaIN. Besides, we add results generated by the model trained without position embedding. Results are shown in Fig. 9. Notice that the resolution during the training stage is 256×256, thus other scales are unseen at training and

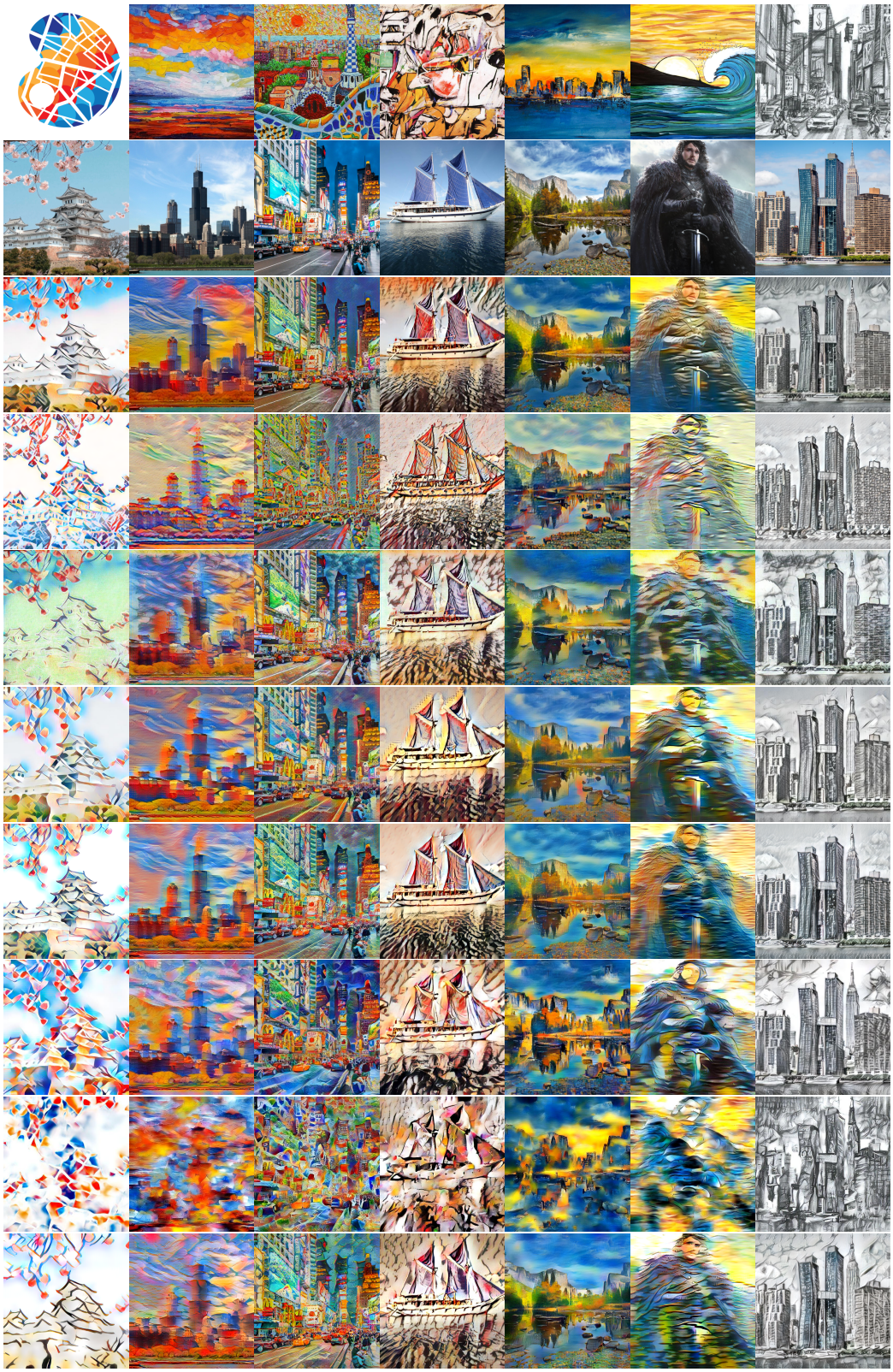
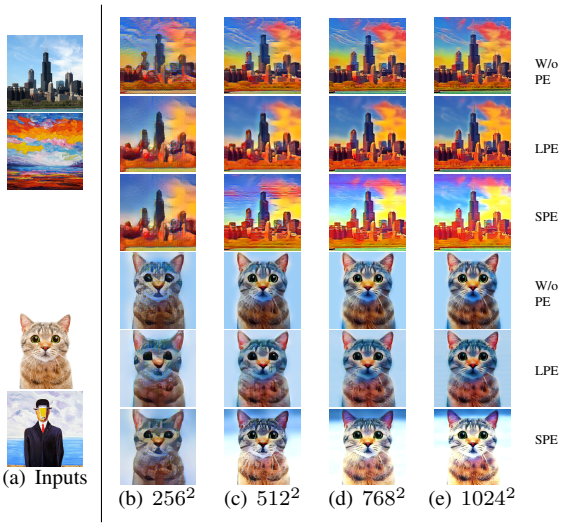


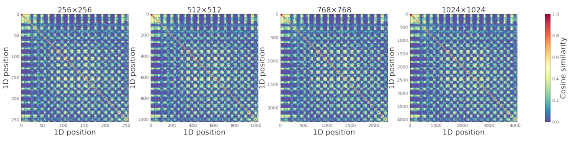
Figure 7. Comparison of visual style transfer results. The first row shows style images, the second row shows content images. The rest rows are stylized results of different methods.

only used for the inference stage. When performing LPE at other resolutions, we use nearest neighbor interpolation on PE to match the input shape. The model with LPE exhibits similar results to the model without any PE, but the one with SPE fails to maintain the consistent style rendering. It binds style to position, also increases the image brightness at other scales.

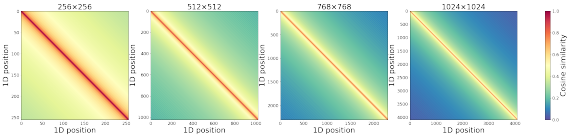


(a) Inputs (b) 256^2 (c) 512^2 (d) 768^2 (e) 1024^2

Figure 9. Visual results of different position embedding setting on Tr-AdaIN.



(a) LPE



(b) SPE

Figure 10. Visualization of position-wise cosine similarity of LPE and SPE on different inference resolutions.

We visualized the position-wise cosine similarity of LPE and SPE on different resolutions, as shown in Fig. 10. When performing LPE on other resolutions, the nearest neighbor interpolation allows the model to perceive the relative position of the image at a larger scale, due to the consistent position embedding value at the same relative position, as shown in Fig. 10(a). When performing SPE at larger resolutions, the SPE at longer distance are computed following sinusoidal position encoding, which are unseen at training stage and differs from the low resolution SPE, as shown in Fig 10(b). The model fails to render style patterns with unseen position information, making the different results at different resolutions.

4.5. Effect of upsampling methods for VST tasks

An additional decode block is necessary to translate the sequence tokens back into images owing to the inconsistent shape. Recently Transformer-based generative models design various decode blocks to invert the tokens into an image. TTSR [38] chooses to stack Transformers, and apply convolution layers to transform the concatenated features back into high-resolution images. IPT [4] introduces multi-tail to deal with different tasks, and each tail uses convolution and pixel shuffle layers to up-sample the features.

We adopt CNNs and MLP into our proposed Transformer auto-encoder model as the upsampling block. The results are shown in Fig.11. Observing that images generated by MLP appear severe checkboard artifact, adjacent patches exhibit inconsistency in perception and produce distinct boundaries. MLP also fails to render complex textures, so that the same pattern appears in each patch. The same observation occurs on both reconstruction and stylization results, implying that directly using MLP is not an appropriate choice for inverting features into images.

(a) I_c (b) I_s (c) Up-CNN (d) Up-MLP

Figure 11. Visual comparisons against CNN and MLP as upsampling decoder. Zoom in for better view.

5. Conclusion

In this paper, we compared CNN and Transformer on the view of style transfer, studied essential elements for designing Transformer-based style transfer models. We adopted three typical visual style transfer algorithms (Tr-NST, Tr-AdaIN, Tr-WCT) into Transformers. We demonstrated that these original CNN-based VST methods don't work on a pre-trained ViT, due to its shape bias both from the learned parameters and the model structure design. By experimenting on setting model variations, we gave the solution for alleviating the strong shape and performing Transformer-based style transfer by retraining our proposed model with additional perceptual loss. We also explored the impact of position embedding and upsampling methods on the stylization results. Using learnable position embedding and not using any position embedding produced similar results, where the sinusoidal positional encoding is not valid due to the learned binding relationship between the style factors

with position information. Moreover, we also demonstrated that combining CNN as the upsampling method is an appropriate choice for avoiding checkboard artifacts and repeated patterns. In the future, we will explore the impact of Transformer design on shape bias and style transfer.

Acknowledgement(s) This work was supported by National Key R&D Program of China under no. 2020AAA0106200, by National Natural Science Foundation of China under nos. 61832016, U20B2070, 6210070958, 62102162, by CASIA-Tencent Youtu joint research project, and by Open Projects Program of National Laboratory of Pattern Recognition.

References

[1] J. An, S. Huang, Y. Song, D. Dou, W. Liu, and J. Luo. Artflow: Unbiased image style transfer via reversible neural flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 862–871, June 2021. DOI:10.1109/cvpr46437.2021.00092. 2, 8

[2] N. Baker, H. Lu, G. Erlikhman, and P. J. Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 14(12):e1006613, 2018. DOI:10.1371/journal.pcbi.1006613. 7

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer International Publishing, 2020. DOI:10.1007/978-3-030-58452-8_13. 2, 3

[4] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12299–12310, June 2021. DOI:10.1109/cvpr46437.2021.01212. 3, 9

[5] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative pretraining from pixels. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 1691–1703, 2020. 2

[6] J.-B. Cordonnier, A. Loukas, and M. Jaggi. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations*, 2020. 3, 5, 7

[7] Y. Deng, F. Tang, W. Dong, H. Huang, C. Ma, and C. Xu. Arbitrary video style transfer via multi-channel correlation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(2):1210–1217, May 2021. DOI:10.1145/3394171.3414015. 1, 2, 8

[8] Y. Deng, F. Tang, W. Dong, W. Sun, F. Huang, and C. Xu. Arbitrary style transfer via multi-adaptation network. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, pages 2719–2727, New York, NY, USA, 2020. Association for Computing Machinery. DOI:10.1145/3394171.3414015. 1, 2

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 3, 7

[10] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016. 4

[11] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. DOI:10.1109/CVPR.2016.265. 1, 2, 4, 6

[12] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 7

[13] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, Jun 2021. DOI:10.1007/s41095-021-0229-5. 2, 3

[14] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1501–1510, Oct 2017. DOI:10.1109/ICCV.2017.167. 1, 2, 5, 6, 8

[15] Y. Jiang, S. Chang, and Z. Wang. TransGAN: Two pure transformers can make one strong GAN, and that can scale up. In *Advances in Neural Information Processing Systems*, 2021. 3

[16] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV 2016*, pages 694–711, Cham, 2016. Springer International Publishing. DOI:10.1007/978-3-319-46475-6_43. 2

[17] N. Kolkin, J. Salavon, and G. Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. DOI:10.1109/CVPR.2019.01029. 1

[18] M. Kumar, D. Weissenborn, and N. Kalchbrenner. Colorization transformer. In *International Conference on Learning Representations*, 2021. 3

[19] K. Lee, H. Chang, L. Jiang, H. Zhang, Z. Tu, and C. Liu. Vitgan: Training gans with vision transformers. *arXiv:2107.04589*, July 2021. https://arxiv.org/abs/2107.04589. 2, 3

[20] X. Li, S. Liu, J. Kautz, and M.-H. Yang. Learning linear transformations for fast image and video style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. DOI:10.1109/cvpr.2019.00393. 2, 8

[21] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, volume 30, 2017. DOI:10.5555/3294771.3294808. 1, 2, 4, 8

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Com-

mon objects in context. In *ECCV 2014*, pages 740–755. Springer International Publishing, 2014. 5

[23] S. Liu, T. Lin, D. He, F. Li, R. Deng, X. Li, E. Ding, and H. Wang. Paint transformer: Feed forward neural painting with stroke prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6598–6607, October 2021. DOI:10.1109/iccv48922.2021.00653. 3

[24] M. Naseer, K. Ranasinghe, S. Khan, M. Hayat, F. Khan, and M.-H. Yang. Intriguing properties of vision transformers. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 2, 3, 7

[25] D. Y. Park and K. H. Lee. Arbitrary style transfer with style-attentional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. DOI:10.1109/cvpr.2019.00603. 2, 7, 8

[26] F. Phillips and B. Mackintosh. Wiki Art Gallery, Inc.: A Case for Critical Thinking. *Issues in Accounting Education*, 26(3):593–608, 08 2011. 5

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. DOI:10.1007/s11263-015-0816-y. 7

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA*, 2015. 4

[29] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou. Training data-efficient image transformers distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, pages 10347–10357, 18–24 Jul 2021. 7

[30] S. Tuli, I. Dasgupta, E. Grant, and T. L. Griffiths. Are convolutional neural networks or transformers more like human vision? *arXiv:2105.07197*, May 2021. https://arxiv.org/abs/2105.07197. 2, 3, 7

[31] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. DOI:10.1109/cvpr.2017.437. 2

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. DOI:10.5555/3295222.3295349. 2, 3, 7

[33] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8741–8750, June 2021. DOI:10.1109/cvpr46437.2021.00863. 3

[34] Z. Wang, L. Zhao, H. Chen, L. Qiu, Q. Mo, S. Lin, W. Xing, and D. Lu. Diversified arbitrary style transfer via deep feature perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. DOI:10.1109/cvpr42600.2020.00781. 2

[35] X. Wu, Z. Hu, L. Sheng, and D. Xu. Styleformer: Real-time arbitrary style transfer via parametric style composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14618–14627, October 2021. DOI:10.1109/iccv48922.2021.01435. 7, 8

[36] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 10524–10533. PMLR, 13–18 Jul 2020. 4

[37] Y. Xu, H. Wei, M. Lin, Y. Deng, K. Sheng, M. Zhang, F. Tang, W. Dong, F. Huang, and C. Xu. Transformers in computational visual media: A survey. *Computational Visual Media*, 8(1):33–62, Mar 2022. DOI:10.1007/s41095-021-0247-3. 3

[38] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. DOI:10.1109/cvpr42600.2020.00583. 2, 9