CLAST: Contrastive Learning for Arbitrary Style Transfer

Xinhao Wang, Wenjing Wang, *Student Member, IEEE*, Shuai Yang, *Member, IEEE*, and Jiaying Liu, *Senior Member, IEEE*

***Abstract*— Arbitrary style transfer aims at migrating the style of a reference style painting to a target content image. Existing methods find it challenging to achieve good content fidelity and style migration at the same time. Moreover, they all rely on manually defined content and style, which is of limited universality and robustness. In this paper, we propose to introduce contrastive learning into style transfer, instructing the network to automatically learn to model the structural content and artistic style based on natural contrastive relationships in style transfer. Compared with existing methods, our learned modeling of content and style is more robust and universal. In addition, we further propose instance-wise contrastive style losses and a patch-wise contrastive content loss to guide style transfer. Combining the proposed contrastive losses and two self-reconstruction strategies, we develop a new style transfer framework, which is pluggable and can be flexibly applied to various style transfer modules. Experimental results demonstrate that our method has strong flexibility and synthesizes stylized images with higher quality.**

***Index Terms*— Style transfer, image synthesis, contrastive learning, image processing, self-supervised learning.**

I. Introduction

ARTISTIC images and paintings are visually attractive and impressive, thus playing an important role in people's daily life. However, creating artistic imagery is labor-consuming and usually takes even experienced artists days of efforts. Nowadays, with the rapid development of the internet and mobile devices, an increasing number of photos and videos are captured and shared.

In order to meet people's growing aesthetic needs, the technique of style transfer is created to automatically transferring a specific artistic style to a photo, benefiting a wide range of users without professional artistic creation skills. As an attractive topic, style transfer has gained increased research attention in recent years and given birth to a great number of industrial products, such as Prisma and DeepArt.

The success of neural networks greatly promotes the development of style transfer. Gatys et al. [1] first proposed to use pretrained image classification networks to encode the content and style, and designed an iterative-optimization-based method: neural style transfer (NST). Later, a series of works have been done for accelerating the transfer procedure [2], improving the generation effect [3], [4], or maintaining temporal consistency for videos [5], [6], [7]. Some subsequent works [7], [8], [9], [10], [11] further enlarge the application range to arbitrary photos and artistic style images, *i.e.*, arbitrary style transfer.

Modeling the abstract concepts of content and style is the core of arbitrary style transfer. Among existing methods, the content similarity between different images is usually measured by the L1 or L2 distance of image classification features [1], some also adopt the modeling based on local self-similarity [3]. As for the modeling of style, many works adopt *global statistics* of pretrained neural networks, such as the traditional Gram-based style loss [1]. *Local patches* can also represent the characteristic of paintings [12]. Nevertheless, these modelings of content and style are all manually defined with pretrained image classification networks, which are counter-intuitive and not flexible enough to capture the extremely diverse content and style features. For example, SANET [9] is one of state-of-the-art arbitrary style transfer methods using the traditional content modeling of [1] and mean-variance-based style modeling. However, as illstruted in Fig. 1(b), SANET destroys the structure of the flowers and generates ghosting human faces due to imbalanced stylization.

In this paper, we revisit and improve style transfer with a new modeling for content and style. Different from existing methods, we propose to learn the representation automatically via the self-supervised learning strategy of contrastive learning [13] that measures the similarities and dissimilarities between sample pairs. Our method comes from how humans intuitively observe artworks as shown in Fig. 1(a). A person may not have exact definitions for content or style, but she/he can judge whether two images match or not. These matching and mismatching relations are natural contrastive relationships, inspiring us to design self-learning style transfer strategies.

On this basis, we propose **contrastive learning for arbitrary style transfer** (**CLAST**). Specifically, the modeling

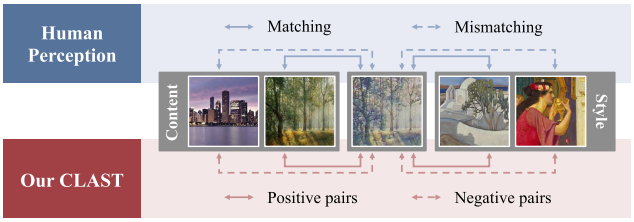
Manuscript received 25 October 2021; revised 8 May 2022; accepted 3 October 2022. Date of publication 25 October 2022; date of current version 31 October 2022. This work was supported in part by the National Natural Science Foundation of China under Contract 62172020; and in part by the Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Guo-Jun Qi. *(Corresponding author: Jiaying Liu.)*

Xinhao Wang, Wenjing Wang, and Jiaying Liu are with the Wangxuan Institute of Computer Technology, Peking University, Beijing 100080, China (e-mail: wxh0510@pku.edu.cn; daooshee@pku.edu.cn; liujiaying@pku.edu.cn).

Shuai Yang is with the S-Lab for Advanced Intelligence, Nanyang Technological University, Singapore 637335 (e-mail: shuai.yang@ntu.edu.sg).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2022.3215899, provided by the authors.

Digital Object Identifier 10.1109/TIP.2022.3215899



(a) Human perception *v.s.* our method

(b) Comparison of our method and SANET [9]

Fig. 1. We imitate human's perception of art by contrastive learning. Benefiting from our appropriate modelings and effective loss designs, our method CLAST outperforms existing methods in terms of content preservation, style migration, and overall visual effects.

of style is learned in a contrastive self-supervised way on painting images alone. The learned modeling is then used to develop instance-wise contrastive style losses. With the help of our robust modeling, the proposed new style losses are naturally in line with people's perception of art, thus effectively guiding the style transfer network to render artistic effects. Meanwhile, we design a patch-wise contrastive content loss and learn the modeling of content during the training of style transfer. Compared with the traditional content loss that employs pretrained and fixed image classification networks, our on-the-fly content encoder can better fit various artistic types. Finally, we develop a pluggable encoder-decoder-based style transfer framework, which can be flexibly combined with various existing style transfer modules.

In conclusion, CLAST has four advantages over previous methods. 1) **Style fidelity**: our data-driven style encoding can render more consistent color and texture; 2) **Content maintenance**: the on-the-fly learned content encoding can better keep the detailed structure of the content image; 3) **Style-content balance**: existing methods are either over-stylized or under-stylized, while our method can adaptively balance the content and style for better overall quality; 4) **Flexibility**: our framework is pluggable and can be applied to various style transfer modules to improve their performance. Code will be released upon publication of the paper.

Our contributions are summarized as follows:

- We propose to model the content and style based on the natural contrastive relationships in style transfer. Compared with previous manually designed style transfer modelings, our representation is automatically learned, and thus is more intuitive, robust and flexible.
- We design a novel training scheme for contrastive arbitrary style transfer. Benefiting from our patch-wise contrastive content loss and instance-wise contrastive style losses, our encoders can not only serve as good feature extractors but also guide balanced style transfer.
- We develop a universal arbitrary style transfer framework CLAST, which can be flexibly applied to various stylization modules. Extensive experimental results show that CLAST achieves superior performance both qualitatively and quantitatively.

The paper is organized as follows. We firstly review existing researches on style transfer, which are divided into three categories according to their modeling of style. After that, in Sec. III, we propose to automatically learn the modeling of content and style based on natural contrastive relationships in style transfer. We further design instance-wise contrastive style losses and a patch-wise contrastive content loss to train the proposed style transfer framework. Later in Sec. IV, we conduct both qualitative and quantitative comparison experiments to demonstrate the superiority of the proposed method. Extensive performance analyses, ablation studies, and applications are also provided. Finally, in Sec. V, we draw conclusions and discuss potential future research directions.

II. Related Work

A. Image Style Transfer

As analyzed in the previous section, the key issue in arbitrary style transfer is how to get an appropriate modeling of content and style. While existing methods mostly adopt the traditional content modeling [1], the style modelings are rather various. Based on the modeling of style, image style transfer algorithms can be divided into three categories: global-statistics-based, local-patch-based, and GAN-based.

1) Global-Statistics-Based Methods: Early traditional methods [14], [15], [16] used the image representations based on hand-crafted low-level features, resulting in weak flexibility and incomplete extraction. The development of convolutional neural networks breaks the limitation of traditional representations, Gatys et al. [1] proposed the pioneering Neural Style Transfer [1], which is the first to utilize a deep neural network to semantically model image style. Neural Style Transfer successfully modeled image style as the correlation of the features in the form of Gram matrix, and further carried out style transfer with an iterative optimization process. This approach, especially the innovative Gram-matrix-based style modeling, inspired other researchers and gave birth to subsequent works.

To avoid the slow optimization procedure, this method is accelerated by [2], [17], which replaced online back-propagation with trainable feed-forward networks in a per-model-per-style mode. [18] further improved structure preservation by designing customized sub-networks.

Inspired by [1], researchers came up with other global statistics to characterize image styles, such as mean, variance [8], and co-variance [19], [20], [21], [22], and proposed the corresponding holistic feature modulation strategies of AdaIN [8] and WCT [19]. In order to better align style features with content features, attention-based feature modulations [9], [23], [24], [25] were designed and achieved promising results. Multi-scale features [26] can provide rich information. Accordingly, many style transfer models adopt multi-layer network architectures [9], [10]. Recently, optimal-transport-based [3] and moment-matching-based [27] modulations were

also proposed. Besides, Kotovenko et al. [28] proposed to optimize strokes rather than pixels to achieve more artistic results.

2) Local-Patch-Based Methods: Image style can be naturally modeled as local patches [29], [30], [31] to simulate the brushstroke and artistic textures in paintings. CNNMRF [12], [32] extracted local patches from the feature maps to model image styles and successfully rendered photo-realistic images. PatchSwap [33] replaced the content feature patches with the best matched style feature patches, but often repetitively used the same patches to cause wash-out artifacts. To solve this problem, Deep Reshuffle [34] softly encouraged the uniform usage of patches. Avatar-Net [10] decorated the image with the style patterns according to the semantic spatial distribution of the content image and applied a multi-scale style transfer, but it usually cannot represent the local and global style patterns at the same time.

In addition, both global-statistics-based and local-patch-based methods use manually defined style modelings. Such modeling is either less flexible to capture the full style information, or less compatible with the content features, resulting in unsatisfactory style patterns or imbalance between the content and style. By comparison, our method learns disentangled and compatible style and content features through contrastive learning and shows promising results.

3) GAN-Based Methods: It is worth noting that Generative Adversarial Networks (GANs) also achieves impressive results in generating images of a certain domain. GAN aims to generate plausible images following the target image distribution, which is suitable to learn images with a certain style. Therefore, solutions based on GAN perform well in collection style transfer, in which the target style is defined by a collection of images, such as Van Gogh's painting collection. CycleGAN [35] proposed an effective cycle consistency constraint to build a pixel-wise relationship between the photos and paintings, and precisely imitated Monet's landscape paintings. CartoonGAN [36] focused on Cartoon and emphasized edge features. Since paintings are often more abstract than photos, making the cycle consistency less available, AST [37] proposed a style-aware content loss to automatically determine how abstract the content feature should be extracted. Besides, feature disentanglement is used in [38], [39], and [40] to extract the compatible content and style features.

Although GAN-based methods render high-quality artistic images, they can only handle one or several fixed artistic types, thus is incapable of arbitrary style transfer. In comparison, our framework can handle any artistic style type, including those unseen during training.

B. Contrastive Learning

The main idea of contrastive learning is to bring positive pairs closer and spread negative pairs apart [41], [42]. By learning what kinds of samples should be viewed as the same and what kinds of samples are different, networks learn to extract discriminative features.

Instead of defining loss functions to directly measure the difference between a model's output and a fixed target, contrastive losses [13] aims to measure the similarities of sample pairs in a representation space. Some recent works focused on training neural networks from scratch [41], [42], others explored to remove the dependency on negative samples [43], [44]. Contrastive learning has many improved versions, such as combining weak and strong augmentation [45], multi-view [46] and contrastive clustering [47]. It also has been widely used in various scenarios, such as object detection [48], domain adaptation [49], [50] and image-to-image translation [51].

Under the common higher-level topic of image generation, CUT [51] discussed to utilize contrastive relationships between two domains as an adjunct to adversarial learning, based on which to further perform image-to-image translation. However, the discussion and method of CUT only considered two domains. By comparison, in the scenario of arbitrary style transfer where each artistic style belongs to its own domain, the input and output images actually involve infinite domains. Therefore, our task is more general and challenging than that of CUT.

III. Contrastive Style Transfer

A. Motivation

The main idea of contrastive learning is to bring positive pairs closer and spread negative pairs apart [41], [42]. By learning what kinds of samples should be viewed as the same and what kinds of samples are different, networks are trained to be effective feature extractors. Contrastive learning has been widely used in various scenarios, such as object detection [48] and image-to-image translation [51]. However, to our knowledge, the high correlation between contrastive learning and style transfer has never been discussed.

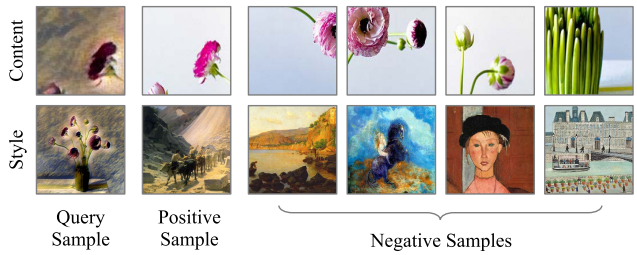
In this paper, we explore the natural contrastive relationships in style transfer. Arbitrary style transfer aims to migrate the artistic patterns from the style image and preserve the content structure of the content image. That is, the stylization result should be artistically close to the style image at the image level, and structurally close to the content image for each local area as shown in Fig. 2(a). We make use of these natural contrastive relationships to learn the style/content modelings and design style transfer frameworks.

B. Contrastive Style Modeling

In arbitrary style transfer, the stylization result is expected to share the same artistic style as the reference style image as shown in Fig. 2(b). Naturally, we come up with the idea of assigning the reference style image as the positive sample and assigning other different artistic images as negative samples. However, before the whole style transfer framework is trained, the stylization result is not readily available. To solve this chicken-and-egg problem, we separate the style feature extraction and style transfer processes, and propose a new contrastive pair design.

Intuitively, two patches randomly cropped from the same painting share almost the same artistic style most of the time. Therefore, we assign them as positive pairs as shown in Fig. 4(a). Since the style of the two patches only differs in rare cases, we consider these mismatching pairs as training

(a) The contrastive relationships in style transfer



(b) Examples of positive and negative selection

Fig. 2. The paradigm of our contrastive style transfer. We use instance-wise contrastive relationships for style and patch-wise contrastive relationships for content. In (b), the five paintings from left to right are: *The Transportation of Marble at Carrara* by Nikolai Ge, *Sorrento* by Nikolai Ge, *Brunhild* (*The Valkyrie*) by Odilon Redon, *Young Girl in Beret* by Amedeo Modigliani, and *Quai aux fleurs* by Louis Vivin. According to wikiart.org, these paintings belong to Romanticism, Romanticism, Symbolism, Expressionism, and Naïve Art (Primitivism).

noises. Experimental results in later sections show that these rare noises hardly affect the effectiveness of our method. With positive and negative samples available, we can now start training a style feature extractor.

Style modeling is learned through the self-supervised task of identifying whether two patches belong to the same image. Specifically, our network consists of a style encoder E_s and a projection head H_s, where the output of H_s is a unit-length vector. The network is trained with painting image data. Considering a query sample S, its positive sample S^+ is randomly cropped from the same painting, while negative samples $S_i^-, i \in \{1, 2, \ldots, N\}$ are patches cropped from other paintings. Note that the negative samples might belong to the same painter or art genre as the positive sample. It is because each image has its own style in arbitrary style transfer. After random cropping, we apply color jittering, discoloration, blurring and flipping with a certain probability. These augmentations encourage the style encoder to extract the color-related and texture-related style at the same time. Otherwise, the samples can be easily classified by simply distinguishing the colors, causing the style encoder to ignore texture information to a certain extent.

Similar to classical contrastive learning, our training objective is as follows:

$$\mathcal{L}_{s\_pre} = \mathbb{E}_S[\mathcal{L}_{\text{NCE}}(H_s(E_s(S)), H_s(E_s(S^+)), H_s(E_s(S^-)))], \quad (1)$$

where $\mathcal{L}_{\text{NCE}}$ stands for the InfoNCE loss [52]:

$$\mathcal{L}_{\text{NCE}}(f, f^+, f^-) = -\log\frac{\sigma(f, f^+)}{\sigma(f, f^+) + \sum_i \sigma(f, f_i^-)}, \quad \sigma(f, g) = \exp(f \cdot g/\tau), \quad (2)$$

and τ is the temperature hyper-parameter. Practically, this is an $(N+1)$-way cross-entropy loss for classifying the positive sample f^+ from other negative samples f^-.

Though the style encoder E_s can well extract both the color-related and texture-related style information, the projection head H_s tends to focus more on semantic information, *i.e.*, texture information according to MoCo [41]. So we train another dual projection head H_{color} which has the identical architecture as H_s but focuses more on colors. Through dividing the style image into 3-by-3 patches and exchange their order, *i.e.*, jigsaw reshuffling, we obtain a structure-destroyed but color-preserved view, noted as S_{jig}. As shown in Fig. 4(b), given a query S_{jig}, its positive sample S_{jig}^+ has the same color appearance but different jigsaw order. The negative samples S_{jig}^- instead have the same jigsaw order as S_{jig}, but their colors are distorted. Under such design, H_{color} is encouraged to utilize more color information to correctly distinguish the positive from the negatives. The training objective for H_{color} is similarly formulated:

$$\mathcal{L}_{color\_pre} = \mathbb{E}_{S_{jig}}[\mathcal{L}_{\text{NCE}}(H_{color}(E_s(S_{jig})), H_{color}(E_s(S_{jig}^+)), H_{color}(E_s(S_{jig}^-)))]. \quad (3)$$

Now we compare our E_s with the widely used classification-pretrained VGG [53]. We collect 40 Monet and Van Gogh paintings, and visualize their style features in Fig. 5. For both the two networks, we use deep features of the *conv*4_1 layer. The features are reshaped to one-dimensional vectors and projected onto a 2D plane by the principal component analysis (PCA) algorithm. While VGG entangles Monet's paintings with Van Gogh's, our E_s clearly distinguishes the paintings of the two artists, indicating that our style modeling can extract style information more comprehensively.

C. Instance-Wise Style Losses

Next, we use the pretrained style feature extractor E_s, H_s and H_{color} to guide style transfer.

During pretraining, for a feature vector of any style sample, the projection heads H_s and H_{color} are trained to bring the feature vector of its positive sample closer, and spread the feature vectors of negative samples apart, where the similarity of features is measured by dot production. Namely, for two style samples x and y, if they have the similar texture-related style, $H_s(E_s(x)) \cdot H_s(E_s(y))$ will be close to 1, otherwise it will be close to 0. The same is true for the color-related style. Therefore, they can be directly used to measure the style difference between two images.

Denoting I_s as the input style image and I_r as the stylization result, we first extract the corresponding style representations $F_s = E_s(I_s)$, $F_r = E_s(I_r)$. Then, we obtain the feature vectors with projection heads:

$$v_s^s = H_s(F_s), \quad v_s^c = H_{color}(F_s), \quad v_r^s = H_s(F_r), \quad v_r^c = H_{color}(F_r).$$

Finally, we minimize the style difference between I_s and I_r during the training of style transfer network:

$$\mathcal{L}_s = \mathbb{E}_{I_s, I_r}[D_s(I_s, I_r)] = \mathbb{E}_{I_s, I_r}[1 - v_s^s \cdot v_r^s], \quad (4)$$

$$\mathcal{L}_{color} = \mathbb{E}_{I_s, I_r}[D_{color}(I_s, I_r)] = \mathbb{E}_{I_s, I_r}[1 - v_s^c \cdot v_r^c], \quad (5)$$

where $\mathcal{L}_s$ and $\mathcal{L}_{color}$ are the instance-wise texture-related and color-related style loss, respectively.

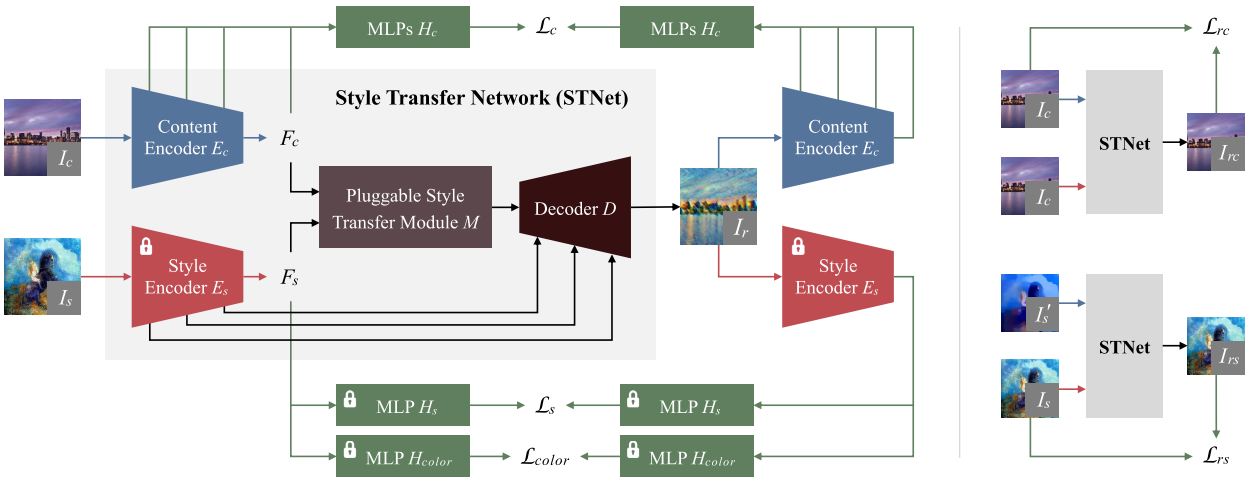
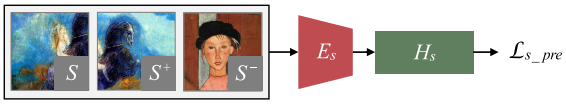
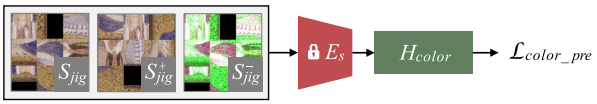


Fig. 3. The overview of our arbitrary style transfer framework. Left: the proposed contrastive style transfer scheme. Right: content identity training (top) and style identity training (bottom). Blue denotes encoding the content, red denotes encoding the style, and green denotes training losses. Locks represent that the parameters are not updated during training.



(a) Training style encoder E_s and projection head H_s



(b) Training color projection head H_{color} with E_s fixed

Fig. 4. The training of contrastive style encoding. Our contrastive style modeling has three network modules: style encoder E_s and two projection heads H_s and H_{color}. We first train E_s and H_s, and then train H_{color} alone with fixed E_s. When training H_{color}, we use jigsaw reshuffling to destroy the content structure of the images.

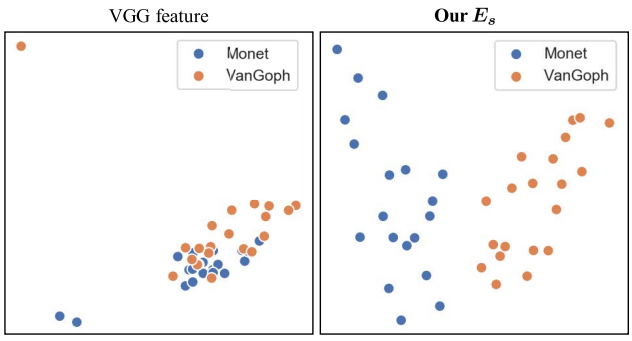
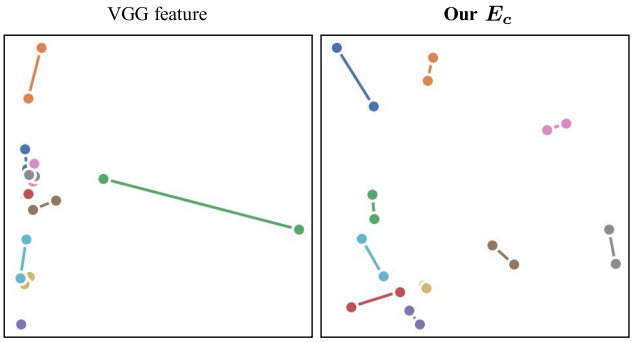
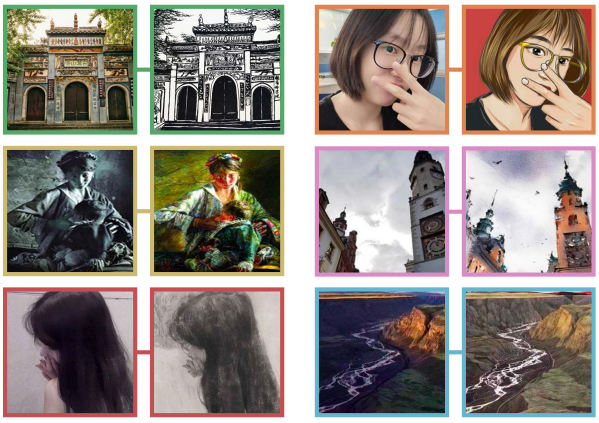


Fig. 5. Style feature visualization results. Blue points indicate Monet's paintings while orange points indicates Van Goph's paintings. While classification-pretrained VGG [53] entangles the feature representations of the two artists, our style encoder E_s clearly distinguishes two styles.



(a) Comparison of the content features.



(b) Examples of the collected photo-painting pairs.

Fig. 6. The feature comparison between our content encoder E_c and the widely used pretrained VGG network. Points of the same color indicate a pair of photo and painting which shares the same content structure. Example photo-painting pairs are shown in (b), where the border of the pairs matches the color of the dots in (a).

Note that different from existing methods [1], [8], [32] which all define the style manually, our modeling of style is automatically learned by simply determining whether or not two images share the same style. Our instance-wise contrastive style losses are also naturally constructed according to the automatic learning process. Therefore, our modeling of style and instance-wise style losses are more robust and can lead to better stylization quality, which is shown in later sections.

D. Contrastive Content Modeling and Patch-Wise Content Loss

In arbitrary style transfer, every patch on the stylization result should be similar in structure to the corresponding patch at the same location on the content image, and be different from other patches as illustrated in Fig. 2(b). Based on this property, we design our contrastive content modeling

and the corresponding patch-wise content loss. During the training of style transfer, even if the stylization result is not properly rendered (for example, at the beginning of training), the content structure can be similar to the content reference image. Therefore, different from style encoding where we first pretrain then fix the style encoder, content encoding can be trained along with the whole style transfer framework.

To perceive the content structure from various scales, we use multi-layer features. Denoting E_c^l as the l-th selected layer of the content encoder, I_c as the content image, and I_r as the stylization result, we randomly choose K_l spatial locations on the feature map from E_c^l, and obtain a stack of features, *i.e.*, the content features of K_l different patches:

$$E_c^l(I_c)_1, E_c^l(I_c)_2, \ldots, E_c^l(I_c)_{K_l},$$
$$E_c^l(I_r)_1, E_c^l(I_r)_2, \ldots, E_c^l(I_r)_{K_l},$$

where $E_c^l(\cdot)_k$ is the content feature of the patch on the k-th location. Similar to style encoding, we then use a projection head H_c^l to map features into an embedding space $\Phi_l^k(I) = H_c^l(E_c^l(I)_k)$. Finally, we maximize the similarity between patches of the same location on I_c and I_r, and reduce the similarity of patches on different locations for I_c. The contrastive content loss is formulated as follows:

$$\mathcal{L}_c = \mathbb{E}_{I_c, I_r, k \neq k', l}[\mathcal{L}_{\text{NCE}}(\Phi_l^k(I_c), \Phi_l^k(I_r), \Phi_l^{k'}(I_c))]. \quad (6)$$

Compared with the traditional content modeling [1] which manually defines the content with pretrained image classifiers, our contrastive modeling is trained on-the-fly with the style transfer framework by simply assigning the natural positive and negative pairs. In this way, the content encoder can perceive content structure under various styles and adaptively learn the modeling of content structure. Therefore, our contrastive modeling can better perceive the content structure across various artistry, bringing better content preservation and content-style trade-off ability.

To compare with the traditional content modeling, we collect 10 photo-painting pairs. Within each pair, the photo and painting share the same content, some of them are shown in Fig. 6(b), others are shown in the supplementary material. The content features extracted by our content encoder E_c and pretrained VGG are visualized in Fig. 6(a). The implementation details are the same as the style feature visualization conducted in Sec. III-B. Our E_c has smaller intra-pair and larger inter-pair distances, demonstrating our modeling of content has a better content extraction ability.

E. Overall Framework

1) Network Architecture: As shown in Fig. 3, our style transfer network (STNet) consists of a content encoder E_c, a style encoder E_s, a pluggable style transfer module M, and a decoder D. Denoting the input content image as I_c, the style image as I_s, we first use E_c and E_s to extract the content and style feature of I_c and I_s, obtaining $F_c = E_c(I_c)$ and $F_s = E_s(I_s)$, respectively. Then, the module M fuses F_c and F_s, where M can be various style transfer modules, including but not limited to AdaIN [8], SANET [9], and dynamic inter-channel filter (DICF) [7]. Finally, the decoder projects the fused representation back to the image domain. In addition, we add multi-level AdaIN skip connections between the style encoder and the decoder. E_s, H_s and H_{color} are not updated during the training of the whole style transfer framework.

Note that the architecture of our sub-networks is replaceable and the selection of module M is arbitrary, our training scheme and contrastive losses can be flexibly applied to various existing style transfer methods.

2) Style Transfer Loss: We use a combination of our contrastive style losses Eqs. (4)-(5), and the contrastive content loss Eq. (6) to guide style transfer:

$$\mathcal{L}_{sty} = \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s + \lambda_{color} \mathcal{L}_{color}, \quad (7)$$

where λ_c, λ_s, and λ_{color} balance different loss functions.

As shown in Fig. 3, two encoders E_c and E_s are used for both generating I_r as parts of STNet and guiding STNet as feature extractors in the loss function. The difference between E_c and E_s is that, E_s together with the two projection heads H_s, H_{color} are first trained on paintings in the style modeling training process with Eq. (1), and then fixed to guide the training of style transfer. In comparison, E_c and its corresponding content projection head H_c is trained along with STNet. Note that most existing methods use a shared pretrained VGG encoder [53] for extracting both the content and style [1], [8], however, sharing encoder may lose domain-specific information and suffer from visual degradation. Our method uses independent encoders instead, which can extract more robust and compatible features.

3) Content Identity: In order to prevent the network from making unexpected changes to the content structure, we introduce a content identity training scheme. Intuitively, we restrict the image to remain unchanged after being stylized by itself. Denoting I_{rc} as the style transfer result of using I_c for both the content and style input, the proposed identity contrastive content loss is:

$$\mathcal{L}_{rc} = \mathbb{E}_{I_c, I_{rc}, k \neq k', l}[\mathcal{L}_{\text{NCE}}(\Phi_l^k(I_c), \Phi_l^k(I_{rc}), \Phi_l^{k'}(I_c))]. \quad (8)$$

4) Style Identity: For better migrating artistic styles, we propose a reconstruction task of restoring textures and colors with the guidance of the style input. Specifically, we first smooth the texture of the painting image I_s by RTV [54], and jitter the color. The style-distorted painting image is denoted as I_s'. Then we use the original painting image I_s to stylize I_s', and expect the stylization result I_{ss} to be identical to I_s. We use our instance-wise contrastive style losses and L2 loss to guide this process:

$$\mathcal{L}_{rs} = \mathbb{E}_{I_s, I_{rs}}[\lambda_{cs}(D_s(I_s, I_{rs}) + D_{color}(I_s, I_{rs})) + \|I_s - I_{rs}\|_2^2], \quad (9)$$

where D_s and D_{color} are the same as in Eqs. (4)-(5).

5) Full Objective: Our final training objective consists of style transfer, content identity, and style identity:

$$\mathcal{L} = \mathcal{L}_{sty} + \lambda_{rc} \mathcal{L}_{rc} + \lambda_{rs} \mathcal{L}_{rs}, \quad (10)$$

where λ_{rc} and λ_{rs} balance different loss functions.

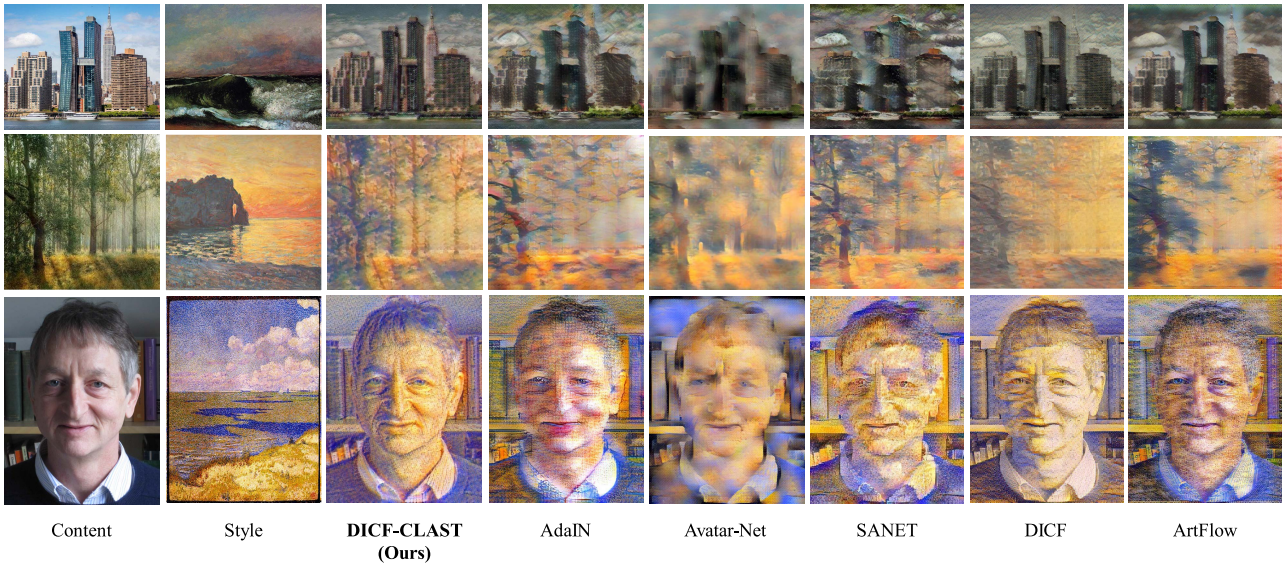


Fig. 7. Comparison results of our DICF-CLAST and state-of-the-art arbitrary style transfer methods. AdaIN [8] fails to render the artistic pattern of the style image. Avatar-Net [10] cannot preserve the detailed structure, making the objects hard to identify. SANET [9] produces appealing style textures but the detailed content structure is distorted. DICF [7] and ArtFlow [11] can better characterize the content details. However, DICF suffers from severe color deviation and ArtFlow cannot completely peel off colors from the content image. By comparison, our method achieves satisfactory content preservation and style fidelity at the same time.

Fig. 8. Comparison of AdaIN-CLAST and AdaIN [8]. Our AdaIN-CLAST renders more vivid style effects and preserves the pivotal structure better (*e.g.*, the lion eyes in row 1).

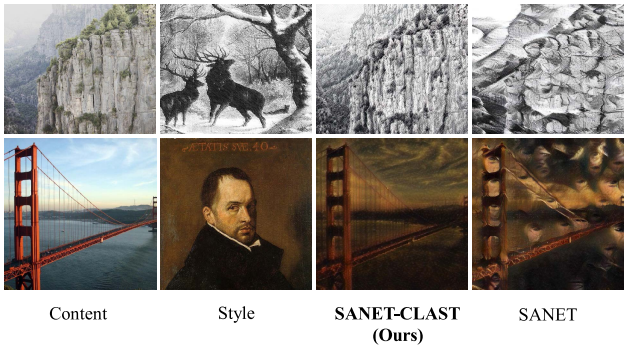


Fig. 9. Comparison of SANET-CLAST and SANET [9]. Our SANET-CLAST eliminates distorted structure and weird repeated patterns, achieves better balance between the content and style.

IV. Experimental Results

A. Implementation Details

To show the flexibility of our method, we apply it to three style transfer modules: AdaIN [8], SANET [9] and DICF [7], resulting in AdaIN-CLAST, SANET-CLAST, and DICF-CLAST, respectively. The networks are trained with MS-COCO [55] as the content and WiKiArt [56] as the style. Style encoding pretraining takes 200 epochs and the whole style transfer network takes 5 epochs. Loss weights are set to: $\lambda_c = 1, \lambda_s = 3, \lambda_{color} = 30, \lambda_{rc} = 1, \lambda_{cs} = 15, \lambda_{rs} = 2$. Note that CLAST can handle any input size and the images we use in experiments are all unseen during training. Please refer to the supplementary material for more detailed training settings and network architectures.

B. Style Transfer Results

We compare our DICF-CLAST with five state-of-the-art arbitrary style transfer methods: AdaIN [8], Avatar-Net [10], SANET [9], DICF [7] and ArtFlow [11]. We also further compare AdaIN, SANET, and DICF with our improved CLAST versions in detail.

1) Visual Effects: Fig. 7 shows the style transfer results of DICF-CLAST and other above-mentioned state-of-the-art algorithms. AdaIN [8] synthesizes the stylized images by simply adjusting the mean and variance of the feature map. Its results are less visually attractive and often retain the colors from the content image to some extent. This is because the traditional content modeling is incapable of completely peeling off colors from the content image. Feature-patch-based Avatar-Net [10] uses a style decorator to make up the content features by semantically aligning style features from an arbitrary style image. However, because of the dependency on patch size, it often fails to migrate both local and global style patterns at the same time. Apart from this, Avatar-Net has a poor performance on content structure preservation (*e.g.*, distorted face in row 3 in Fig. 7). SANET [9] applies a style attention mechanism to align style features with the content features. Though it can produce appealing style

TABLE I

QUANTITATIVE EVALUATION RESULTS. RED AND BLUE INDICATE THE BEST AND SECOND-BEST PERFORMANCE

Method	Ours			AdaIN [8]	AvatarNet [10]	SANET [9]	DICF [7]	ArtFlow [11]
	AdaIN-CLAST	SANET-CLAST	DICF-CLAST					
SSIM ↑	0.625	0.580	0.626	0.373	0.458	0.438	0.566	0.607
SSIM rank ↑	6.300	5.232	6.358	1.482	3.272	2.962	4.640	5.754
EMD ↓	0.612	0.594	0.598	0.812	0.618	0.708	0.638	0.663
EMD rank ↓	4.060	3.840	3.756	5.998	4.230	5.182	4.320	4.614

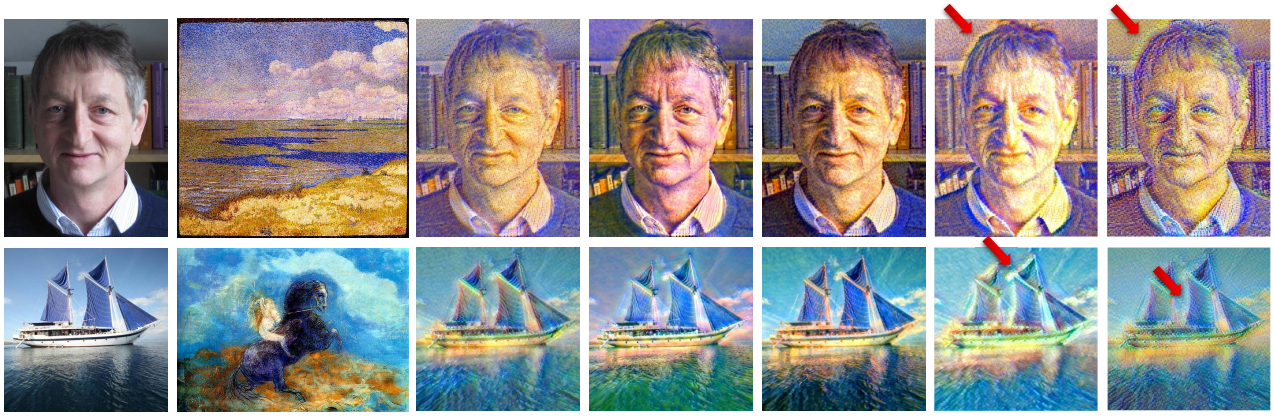


Fig. 10. Ablation studies on the effect of each loss in our contrastive style transfer, where w/o $\mathcal{L}_s$, w/o $\mathcal{L}_{color}$, w/o $\mathcal{L}_{rs}$ and w/o $\mathcal{L}_{rc}$ indicate removing the instance-wise texture-related style loss, instance-wise color-related style loss, style identity loss and content identity loss, respectively.

TABLE II

QUANTITATIVE EVALUATION RESULTS FOR ABLATION STUDIES

Method	**DICF-CLAST (Full Version)**	w/o $\mathcal{L}_s$	w/o $\mathcal{L}_{color}$	w/o $\mathcal{L}_{rs}$	w/o $\mathcal{L}_{rc}$
SSIM ↑	0.627	0.692	0.712	0.591	0.572
SSIM rank ↑	2.680	4.250	4.490	1.887	1.693
EMD ↓	0.616	0.833	0.861	0.751	0.659
EMD rank ↓	1.893	2.880	2.920	2.443	1.757

Fig. 11. Comparison of DICF-CLAST and DICF [7]. Our DICF-CLAST notably improves the color fidelity and overall visual quality.

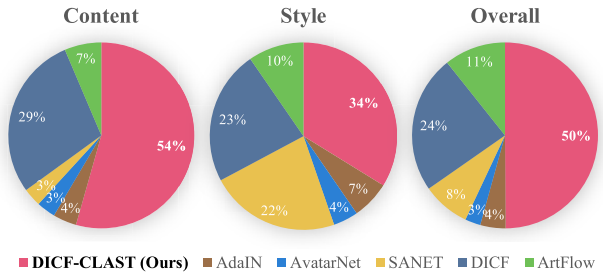


Fig. 12. User preference benchmarking results. Our DICF-CLAST outperforms other state-of-the-art methods in all the three aspects.

textures most of the time, the detailed content structure is often distorted (*e.g.*, row 1, 3 in Fig. 7), and repeated patterns occur occasionally. DICF [7] can well characterize content details, but it sometimes suffers from severe color deviation compared with corresponding style images (*e.g.*, row 1, 2 in Fig. 7). ArtFlow [11] adopts a projection flow network and performs unbiased image style transfer, so it preserves the content structure well. However, it cannot completely peel off colors from the content image, remaining the color of the black clothes in the third row.

Compared with these arbitrary style transfer methods, DICF-CLAST achieves both pleasant structure preservation and faithful style effects, demonstrating the effectiveness of our contrastive style transfer scheme.

We show a more detailed comparison between AdaIN-CLAST, SANET-CLAST, DICF-CLAST and their original

Fig. 13. Comparison of style transfer results with NST. The result synthesized with our style encoder E_s has a better color distribution and more vivid textures.

TABLE III
EXECUTION TIME COMPARISON (IN SECONDS)

Method	Time
AdaIN [8]	0.1158
AvatarNet [10]	0.4669
SANET [9]	0.2543
DICF [7]	0.2139
ArtFlow [11]	0.6560
AdaIN-CLAST (ours)	0.1263
SANET-CLAST (ours)	0.1389
DICF-CLAST (ours)	0.2473

versions in Figs. 8, 9, and 11. For AdaIN, CLAST can render more vivid style effects and better preserve shapes and edges. For SANET, CLAST eliminates distorted structure (*e.g.*, unexpected eyes in row 2 in Fig. 9) and better balances the content and style. For DICF, with CLAST, the colors are more in line with the style image. In summary, our method can improve the visual quality of stylization results to a large extent.

2) Quantitative Comparison: Inspired by [21], we use the Structural Similarity Index (SSIM) between original content images and stylized images to measure the performance of content preservation. Also, we use the Earth Mover's Distance (EMD) between the color histograms of original style images and stylized images to measure the performance of color migration. For each method, we use 20 content images and 15 style images to generate 300 results. We compute their average SSIM, EMD, and ranks. As Table I shows, for each metric, both the best and the second are our methods, demonstrating the superiority of our method.

3) User Study: To evaluate human-eye visual effects, we conduct a user study on Amazon Mechanical Turk. The participants are requested to choose the best result among a set of candidates in perspectives of content preservation, style migration, and overall quality, respectively. We use 10 content images and 20 style paintings to generate 200 results for each method. Each content-style pair is assigned to 5 different participants, summing up to 3,000 votes in total.

We show the percentage of votes for each method in Fig. 12. The results demonstrate that our method outperforms other methods in all three aspects.

C. Ablation Studies

We present the results of DICF-CLAST trained by various loss combinations in Fig. 10, please zoom in to see details. Discarding our instance-wise texture-related style loss $\mathcal{L}_s$, the model fails to render the noise-like texture of the style

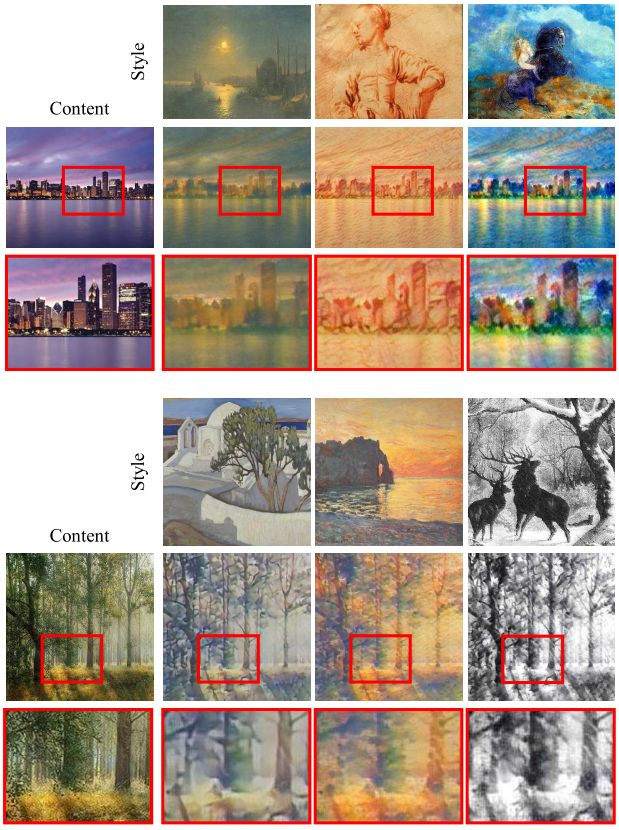


Fig. 14. Our stylization results synthesized from the same content image and different artistic paintings. Our method can adaptively balance the content and style according to the artistic painting.

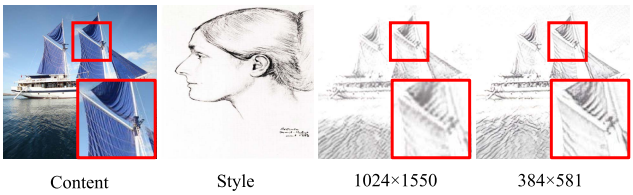


Fig. 15. Our method generates blurry textures when the resolution of the input style image is too high. This problem can be solved by manually down-sampling the style image.

image compared with our full version. Besides, the colors also degrade. Removing the instance-wise color-related style loss $\mathcal{L}_{color}$, the overall color distribution deviates obviously. Abandoning the style reconstruction loss $\mathcal{L}_{rs}$ harms both the maintenance of content structure and the migration of style patterns. Without the content reconstruction loss $\mathcal{L}_{rc}$, the model cannot well preserve the detailed content structure. Obvious ringing artifacts emerge as indicated by the red arrows. In comparison, our full style transfer version achieves

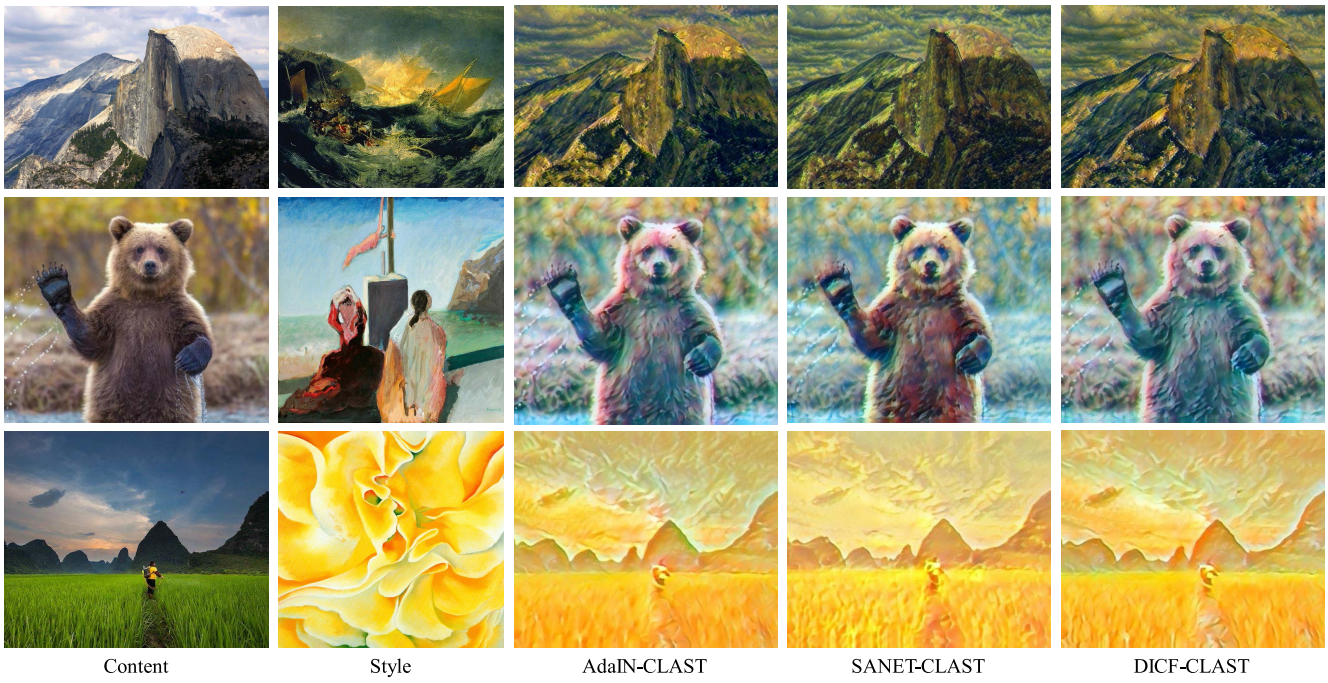


Fig. 16. Comparison results of three versions of our method, AdaIN-CLAST, SANET-CLAST and DICF-CLAST.

the best style transfer effect, demonstrating the effectiveness of our loss designs.

Similar to the quantitative comparison conducted in Sec. IV-B2, we additionally calculate the average SSIM, EMD and ranks of each version. The quantitative results are shown in Table II. Deleting our instance-wise contrastive style losses results in terrible performance of color migration. Notice that the average EMD of the version without $\mathcal{L}_{color}$ is higher than that of the full version, which further verifies that our instance-wise color-related style loss $\mathcal{L}_{color}$ can help with color rendering. Meanwhile, chucking the style reconstruction loss $\mathcal{L}_{rs}$ leads to both worse SSIM and EMD. Dropping the content reconstruction loss $\mathcal{L}_{rc}$ significantly harms the maintenance of the content structure. In summary, the quantitative results also confirm that each of our losses plays their own role.

D. Performance Analysis

1) Style Encoding: Most style transfer methods utilize the pretrained VGG [53] to extract the style features. By comparison, exploring the natural contrastive relationships in style transfer, our style encoder and projection heads automatically learn robust and flexible style modelings. To demonstrate that our style encoder can extract better artistic representations, we conduct the optimization-based NST [1] with pretrained VGG or our style encoder separately. We show the results in Fig. 13. Compared with VGG, the result of our style encoder E_s not only better eliminates colors from the content image but also contains more vivid textures.

In Sec. III-B, we apply color jittering and discoloration to encourage the style encoder to extract both the color-related and texture-related style. Otherwise, the samples can be easily classified by simply distinguishing the colors, causing the style encoder to ignore some texture information. To demonstrate this with NST, we train another style encoder E_s' without the above mentioned step. As shown in Fig. 13, the result of E_s' has fewer textures and worse overall quality.

2) Content-Style Balance: Our modelings of content and style are automatically learned with natural contrastive pairs, therefore, our method can adaptively balance the content and style. As shown in Fig. 14, our method can adjust the content and style satisfactorily and synthesize vivid results. For example, in the first three rows of Fig. 14, for the flat style (2nd column), our result retains the clear buildings and smooth sea surface; for the style with rich textures (3rd column), our model renders vivid textures to match the strokes in the style image.

3) Influence of the Style Transfer Module: We compare the style transfer results synthesized by AdaIN-CLAST, SANET-CLAST and DICF-CLAST in Fig. 16. Their only difference is the selection of the style transfer module M. In the 1st and 3rd row of Fig. 16, since DICF [7] is a newly-proposed and powerful module, DICF-CLAST renders more vivid style effects and more matching colors (*e.g.*, in row 1, the sky synthesized by AdaIN-CLAST retains some blue color, while the stylization result produced by SANET-CLAST is less appropriate on colors), the result is also more appealing from the aspect of overall visual quality. However, for the style of the 2nd row, due to the attention mechanism in the SANET [9] module, only SANET-CLAST captures the dark wine-red from the style image and generates the best result.

In summary, our method has strengths and weaknesses based on the properties of the selected style transfer module. The conclusion is also in line with the quantitative comparison results conducted in Sec IV-B2. The style transfer module

should be chosen according to both the specific application scenario and the specialties of style transfer modules.

E. Runtime Analysis

Tab. III shows the runtime performance of our methods and other state-of-the-art arbitrary style transfer methods. We calculate the execution time by averaging the time to generate 100 stylized images. The input content and style images are all 512 × 512 resolution. It is clear that after applying our CLAST, AdaIN-CLAST and DICF-CLAST remain almost the same efficiency as the original algorithms. Particularly, our SANET-CLAST is nearly 2 times faster than SANET [9]. Our method can feasibly process style transfer in real time with various style transfer modules.

F. Limitation

Since the receptive field of the style encoder is limited, our method does not perform well on high-resolution style images. In Fig. 15, the resolution of the input style image is 1024 × 1550. Accordingly, the texture of our result is blurry. This problem can be easily solved by down-sampling the style image. By automatically resizing the shortest edge to 384, the pencil drawing texture becomes vivid and obvious.

Another limitation is that, although our method achieves superior performance, the training process is slightly complicated and requires large-scale training datasets. In the future, we will explore how to get rid of the dependence on pre-training processes and large-scale training datasets, such as zero-shot learning [57].

V. Conclusion

In this paper, we introduce contrastive learning into arbitrary style transfer. We point out that the style can be determined through comparing distinct painting instances, while the content is the patch-level correspondence between the style transfer result and the content image. On this basis, we construct the style and content encoders and propose a new framework for style transfer. Our pluggable losses and training strategies can be applied to various style transfer methods. Experimental results demonstrate not only the effectiveness, but also the superiority and flexibility of our new framework and loss designs. We believe that our work can inspire subsequent style transfer algorithms.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.

[2] J. Johnson, A. Alahi, and F. F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.

[3] N. Kolkin, J. Salavon, and G. Shakhnarovich, "Style transfer by relaxed optimal transport and self-similarity," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10051–10060.

[4] E. Heitz, K. Vanhoey, T. Chambon, and L. Belcour, "A sliced Wasserstein loss for neural texture synthesis," 2020, *arXiv:2006.07229*.

[5] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," in *Proc. German Conf. Pattern Recognit.*, B. Rosenhahn and B. Andres, Eds., 2016, pp. 26–36.

[6] W. Wang, J. Xu, L. Zhang, Y. Wang, and J. Liu, "Consistent video style transfer via compound regularization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12233–12240.

[7] W. Wang, S. Yang, J. Xu, and J. Liu, "Consistent video style transfer via relaxation and regularization," *IEEE Trans. Image Process.*, vol. 29, pp. 9125–9139, 2020.

[8] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1501–1510.

[9] D. Y. Park and K. H. Lee, "Arbitrary style transfer with style-attentional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5880–5888.

[10] L. Sheng, Z. Lin, J. Shao, and X. Wang, "Avatar-Net: Multi-scale zero-shot style transfer by feature decoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8242–8250.

[11] J. An, S. Huang, Y. Song, D. Dou, W. Liu, and J. Luo, "ArtFlow: Unbiased image style transfer via reversible neural flows," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 862–871.

[12] C. Li and M. Wand, "Combining Markov random fields and convolutional neural networks for image synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2479–2486.

[13] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 1735–1742.

[14] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proc. 22nd Annu. Conf. Comput. Graph. Interact. Techn.*, 1995, pp. 229–238.

[15] B. Julesz, "Visual pattern discrimination," *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp. 84–92, Feb. 1962.

[16] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 49–70, Oct. 2000.

[17] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *Proc. CVPR*, Jul. 2017, pp. 6924–6932.

[18] M.-M. Cheng, X.-C. Liu, J. Wang, S.-P. Lu, Y.-K. Lai, and P. L. Rosin, "Structure-preserving neural style transfer," *IEEE Trans. Image Process.*, vol. 29, pp. 909–920, 2020.

[19] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[20] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz, "A closed-form solution to photorealistic image stylization," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 453–468.

[21] J. Yoo, Y. Uh, S. Chun, B. Kang, and J.-W. Ha, "Photorealistic style transfer via wavelet transforms," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9036–9045.

[22] Z. Wang et al., "Diversified arbitrary style transfer via deep feature perturbation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7789–7798.

[23] Y. Deng, F. Tang, W. Dong, W. Sun, F. Huang, and C. Xu, "Arbitrary style transfer via multi-adaptation network," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2719–2727.

[24] S. Liu et al., "AdaAttN: Revisit attention mechanism in arbitrary neural style transfer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6649–6658.

[25] X. Wu, Z. Hu, L. Sheng, and D. Xu, "StyleFormer: Real-time arbitrary style transfer via parametric style composition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14618–14627.

[26] L. Zhu, H. Fan, Y. Luo, M. Xu, and Y. Yang, "Temporal cross-layer correlation mining for action recognition," *IEEE Trans. Multimedia*, vol. 24, pp. 668–676, 2022.

[27] N. Kalischek, J. D. Wegner, and K. Schindler, "In the light of feature distributions: Moment matching for neural style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9382–9391.

[28] D. Kotovenko, M. Wright, A. Heimbrecht, and B. Ommer, "Rethinking style transfer: From pixels to parameterized brushstrokes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12196–12205.

[29] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 1033–1038.

[30] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.*, 2001, pp. 341–346.

[31] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proc. ACM SIGGPRAH*, 2001, pp. 327–340.

[32] C. Li and M. Wand, "Precomputed real-time texture synthesis with Markovian generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 702–716.

[33] T. Q. Chen and M. Schmidt, "Fast patch-based style transfer of arbitrary style," 2016, *arXiv:1612.04337*.

[34] S. Gu, C. Chen, J. Liao, and L. Yuan, "Arbitrary style transfer with deep feature reshuffle," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8222–8231.

[35] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[36] Y. Chen, Y.-K. Lai, and Y.-J. Liu, "CartoonGAN: Generative adversarial networks for photo cartoonization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9465–9474.

[37] A. Sanakoyeu, D. Kotovenko, S. Lang, and B. Ommer, "A style-aware content loss for real-time HD style transfer," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 698–714.

[38] D. Kotovenko, A. Sanakoyeu, P. Ma, S. Lang, and B. Ommer, "A content transformation block for image style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10032–10041.

[39] D. Kotovenko, A. Sanakoyeu, S. Lang, and B. Ommer, "Content and style disentanglement for artistic style transfer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4422–4431.

[40] J. Svoboda, A. Anoosheh, C. Osendorfer, and J. Masci, "Two-stage peer-regularized feature recombination for arbitrary image style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13816–13825.

[41] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[42] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. IEEE Int. Conf. Mach. Learn.*, Nov. 2020, pp. 1597–1607.

[43] J.-B. Grill et al., "Bootstrap your own latent: A new approach to self-supervised learning," 2020, *arXiv:2006.07733*.

[44] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15750–15758.

[45] X. Wang and G.-J. Qi, "Contrastive learning with stronger augmentations," 2021, *arXiv:2104.07713*.

[46] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, "COMPLETER: Incomplete multi-view clustering via contrastive prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11174–11183.

[47] Y. Li, P. Hu, J. Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 8547–8555.

[48] E. Xie et al., "DetCo: Unsupervised contrastive learning for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8392–8401.

[49] W. Wang, W. Yang, and J. Liu, "HLA-face: Joint high-low adaptation for low light face detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16195–16204.

[50] R. Wang, Z. Wu, Z. Weng, J. Chen, G.-J. Qi, and Y.-G. Jiang, "Cross-domain contrastive learning for unsupervised domain adaptation," *IEEE Trans. Multimedia*, early access, Jan. 27, 2022.

[51] T. Park, A. A. Efros, R. Zhang, and J. Zhu, "Contrastive learning for unpaired image-to-image translation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 319–345.

[52] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[54] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 139:1–139:10, 2012.

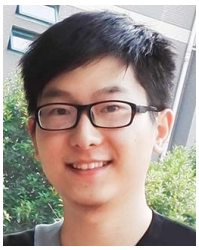
[55] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[56] F. Phillips and B. Mackintosh, "Wiki art gallery, Inc.: A case for critical thinking," *Issues Accounting Educ.*, vol. 26, no. 3, pp. 593–608, 2011.

[57] B. Li, Y. Gou, J. Z. Liu, H. Zhu, J. T. Zhou, and X. Peng, "Zero-shot image dehazing," *IEEE Trans. Image Process.*, vol. 29, pp. 8457–8466, 2020.

Xinhao Wang received the B.S. degree in intelligence science from Peking University, Beijing, China. His current research interests include style transfer and deep learning.

Wenjing Wang (Student Member, IEEE) received the B.S. degree in data science from Peking University, Beijing, China, in 2019, where she is currently pursuing the Ph.D. degree with the Wangxuan Institute of Computer Technology. Her current research interests include image enhancement, image synthesis, and deep learning.



Shuai Yang (Member, IEEE) received the B.S. and Ph.D. degrees (Hons.) in computer science from Peking University, Beijing, China, in 2015 and 2020, respectively. He was a Visiting Scholar with Texas A&M University, from September 2018 to September 2019. He was a Visiting Student with the National Institute of Informatics, Japan, from March 2017 to August 2017. He is currently a Postdoctoral Research Fellow with the NTU AI Corporate Laboratory, Nanyang Technological University. His current research interests include image stylization and image generation. He received the IEEE ICME 2020 Best Paper Awards and the IEEE MMSP 2015 Top10% Paper Awards.

Jiaying Liu (Senior Member, IEEE) received the Ph.D. degree (Hons.) in computer science from Peking University, Beijing, China, in 2010.

She was a Visiting Scholar with the University of Southern California, Los Angeles, CA, USA, from 2007 to 2008. She was a Visiting Researcher with Microsoft Research Asia, in 2015, supported by the Star Track Young Faculties Award. She is currently an Associate Professor and a Boya Young Fellow with the Wangxuan Institute of Computer Technology, Peking University. She has authored more than 100 technical articles in refereed journals and proceedings, and holds 70 granted patents. Her current research interests include multimedia signal processing, compression, and computer vision. She is a Senior Member of CSIG and a Distinguished Member of CCF. She has served as a member for the Multimedia Systems and Applications Technical Committee (MSA TC) and the Visual Signal Processing and Communications Technical Committee (VSPC TC) in IEEE Circuits and Systems Society. She received the IEEE ICME 2020 Best Paper Award and the IEEE MMSP 2015 Top10% Paper Award. She has served as the Technical Program Chair for the IEEE ICME-2021/ACM ICMR-2021, the Area Chair for CVPR-2021/ECCV-2020/ICCV-2019, and the CAS Representative at the ICME Steering Committee. She was the APSIPA Distinguished Lecturer (2016–2017). She has also served as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUIT SYSTEM FOR VIDEO TECHNOLOGY, and the *Journal of Visual Communication and Image Representation*.