Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
Special Track on AI, the Arts and Creativity

Universal Video Style Transfer via Crystallization, Separation, and Blending

Haofei Lu[1] and **Zhizhong Wang**[*2]
[1]School of Computer Science and Engineer, Southeast University
[2]College of Computer Science and Technology, Zhejiang University
josh00@seu.edu.cn, endywon@zju.edu.cn

Abstract

Universal video style transfer aims to migrate arbitrary styles to input videos. However, how to maintain the temporal consistency of videos while achieving high-quality arbitrary style transfer is still a hard nut to crack. To resolve this dilemma, in this paper, we propose the *CSBNet* which involves three key modules: 1) the **Crystallization (Cr) Module** that generates several orthogonal crystal nuclei, representing hierarchical stability-aware content and style components, from raw VGG features; 2) the **Separation (Sp) Module** that separates these crystal nuclei to generate the stability-enhanced content and style features; 3) the **Blending (Bd) Module** to cross-blend these stability-enhanced content and style features, producing more stable and higher-quality stylized videos. Moreover, we also introduce a new pair of component enhancement losses to improve network performance. Extensive qualitative and quantitative experiments are conducted to demonstrate the effectiveness and superiority of our CSBNet. Compared with the state-of-the-art models, it not only produces temporally more consistent and stable results for arbitrary videos but also achieves higher-quality stylizations for arbitrary images.

1 Introduction

Style transfer aims to render a content target with the desired style of a given style reference. Recently, the seminal work of [Gatys *et al.*, 2016] showed that the Gram matrix (*i.e.*, feature correlation) of features extracted from a Deep Convolutional Neural Network (DCNN) could represent the style patterns well. Since then, significant efforts have been made to improve the performance in various aspects, including efficiency [Johnson *et al.*, 2016], quality [Wang *et al.*, 2021; Chen *et al.*, 2021b], generality [Huang and Belongie, 2017; Li *et al.*, 2017], and diversity [Ulyanov *et al.*, 2017; Wang *et al.*, 2020b], *etc*. However, existing style transfer approaches are mostly designed for still images, while the stylization for dynamic videos is still an open challenge.

*Corresponding author.

Compared with image style transfer, video style transfer is a much more challenging task. The method should consider both the temporal consistency and stylization quality of the stylized videos. Existing image style transfer methods, while achieving satisfactory results for still images, often result in significant jitters and flickers when applied to videos due to the lack of in-depth consideration into video temporal stability. To overcome this problem, [Ruder *et al.*, 2016] introduced optical flow prediction to [Gatys *et al.*, 2016] to maintain the video-frame consistency. However, it takes too much time to stylize a single frame. For fast video style transfer, [Chen *et al.*, 2017] designed a feed-forward optical flow and mask estimation network, and [Gao *et al.*, 2020] proposed a multi-style framework that estimates the light flow and uses temporal constraint. These methods benefit from optical flow prediction to maintain temporal stability but rely highly on estimation accuracy. What's more, works like [Wang *et al.*, 2020a] have justified that optical flow not only suppresses the stylization quality but also may bring obvious video jelly effects in the results.

Recently, [Li *et al.*, 2019] proposed a stable linear transformation that is suitable for frame-based video style transfer. Later, [Wang *et al.*, 2020a] introduced a novel interpretation of temporal consistency without optical flow estimation and achieved consistent zero-shot video style transfer. Based on it, [Deng *et al.*, 2021] proposed multi-channel correlation for arbitrary video style transfer. These approaches achieved the state-of-the-art video style transfer. However, their stylization quality is not satisfactory, and some slight brushstroke jitters still exist. Therefore, how to maintain the temporal consistency of videos while achieving high-quality arbitrary style transfer is still a hard nut to crack.

In order to avoid using the optical flow method and complete high-quality arbitrary style transfer while maintaining the video continuity, in this paper, we propose a novel frame-based network for universal video style transfer. Our method, termed *CSBNet*, consists of three key modules: 1) the **Crystallization (Cr) Module** that uses singular value decomposition to generate several orthogonal matrices called crystal nuclei, representing hierarchical stability-aware content and style components, from raw VGG features; 2) the **Separation (Sp) Module** that separates these crystal nuclei to generate the stability-enhanced content and style features; 3) the **Blending (Bd) Module** to cross-blend these stability-

enhanced content and style features, producing more stable and higher-quality stylized videos. At the same time, we also introduce a new pair of component enhancement losses to improve network performance. Extensive qualitative and quantitative experiments demonstrate that our method can effectively suppress the generation of local artifacts and inherently maintain the smoothness of input videos while successfully keeping the content structures and style details, significantly outperforming the state-of-the-art (SOTA) models.

The main contributions of our method are:

- A novel frame-based network *CSBNet* consisting of a Crystallization (Cr) Module, a Separation (Sp) Module, and a Blending (Bd) Module, to achieve consistent universal video style transfer with higher stylization quality.
- A new pair of component enhancement losses to improve the performance of our network, helping produce more stable and higher-quality results.
- Extensive qualitative and quantitative experiments to demonstrate the effectiveness and superiority of our method in both image and video style transfer.

2 Related Work

Image Style Transfer. [Gatys *et al.*, 2016] firstly introduced DCNN to artistic style transfer. In their method, the style of an image is represented by the Gram matrix (*i.e.*, feature correlation) of the intermediate features extracted from DCNN. While achieving visually stunning results, the iterative procedure is time-consuming, and the output is sensitive to input noise. Later, [Li and Wand, 2016] improved the stability by introducing Markov Random Fields (MRFs) and proposed a patch-based method. To achieve real-time transfer, [Johnson *et al.*, 2016] introduced a feed-forward approach by training feed-forward networks. However, the trained networks are fixed for pre-defined styles. To achieve arbitrary style transfer, AdaIN [Huang and Belongie, 2017] aligned the feature maps of content images with style ones by normalizing means and variances. Also, WCT [Li *et al.*, 2017] orthogonalized the content features and matched the covariance matrix of the style features. However, outputs generated by AdaIN and WCT failed to represent the detailed textures sufficiently and unexpectedly brought twisted style patterns at some local areas more or less. Further, Linear [Li *et al.*, 2019] learned a linear transformation to better preserve content affinity during style transfer. SANet [Park and Lee, 2019] employed a style-attention module to better capture local style patterns. AAMS [Yao *et al.*, 2019] introduced self-attention methods to improve output quality. AdaAttN [Liu *et al.*, 2021] proposed a novel attention and normalization module for arbitrary style transfer. IECAST [Chen *et al.*, 2021a] incorporated the adversarial nets and contrastive losses to improve SANet.

Despite the recent progress in image style transfer, few of them considered the temporal consistency, resulting in obvious flickering effects when applied to videos. Moreover, there's still improvement room for existing style transfer approaches to better blend content structures with style patterns to generate higher-quality stylized results.

Video Style Transfer. Current video style transfer methods can be roughly divided into two categories: 1) methods based on optical flow prediction and 2) non-optical flow prediction methods with single frame constraints.

In order to maintain video-frame consistency, [Ruder *et al.*, 2016] introduced optical flow prediction of the previous frame to guide the generation of the current time. However, it needs several minutes to generate a single output frame, which is unsuitable for large-scale applications. Consequently, [Chen *et al.*, 2017] designed a feed-forward network for fast video style transfer. [Gao *et al.*, 2020] introduced ConvLSTMs to stabilize results and FlowNet(s) to simulate optical-flow prediction. These methods benefit from optical flow prediction to maintain temporal consistency but rely highly on estimation accuracy. In addition, optical-flow guidance also limits the quality of stylization [Wang *et al.*, 2020a].

Recently, Linear [Li *et al.*, 2019] learned a stable linear transformation that is suitable for frame-based video style transfer. ReReVST [Wang *et al.*, 2020a] introduced a novel interpretation of temporal consistency without optical flow estimation and achieved consistent zero-shot video style transfer via relaxation and regularization. MCCNet [Deng *et al.*, 2021] proposed multi-channel correlation for arbitrary video style transfer. These methods achieved the state-of-the-art video style transfer. However, their stylization quality is unsatisfactory, and some slight brushstroke jitters still exist.

To avoid using the optical flow estimation while obtaining high-quality stylization and maintaining the video continuity, we propose a novel frame-based universal video style transfer approach via crystallization, separation, and blending. The stability-aware content and style components of the content and style images are first crystallized and separated. Then, the stability-enhanced content feature of the content image and the style feature of the style image are cross-blended, thus producing temporally more stable and higher-quality results.

3 Proposed Method

3.1 Network Structure

We first depict the main structure of our network from the high level, which is based on the encoder-decoder model. As shown in Fig. 1, CSBNet uses a pre-trained VGG-19 network [Simonyan and Zisserman, 2014] as the encoder and the symmetric network as the decoder. Given a content image I_c and a style image I_s, we first generate their corresponding VGG features $F_c = \mathrm{VGG}(I_c)$ and $F_s = \mathrm{VGG}(I_s)$.

Then, the VGG features F_c and F_s are fed into the **Crystallization (Cr) Module** (Sec. 3.2) to generate several stability-aware orthogonal matrices $A_1, A_2, ..., A_N$, namely crystal nuclei in our work:

$$A_1, A_2, ..., A_N = \mathrm{Cr}(F), \quad F \in \{F_c, F_s\}. \tag{1}$$

Next, the **Separation (Sp) Module** (Sec. 3.2) completes the separation of these crystal nuclei according to the empirically determined cursors K_c and K_s (which control the degree of separation), generating the stability-aware content component $F^{content}$ and style component F^{style}:

$$\begin{aligned} F_c^{content}, F_c^{style} &= \mathrm{Sp}_1(\mathrm{Cr}(F_c), K_c), \\ F_s^{content}, F_s^{style} &= \mathrm{Sp}_2(\mathrm{Cr}(F_s), K_s). \end{aligned} \tag{2}$$

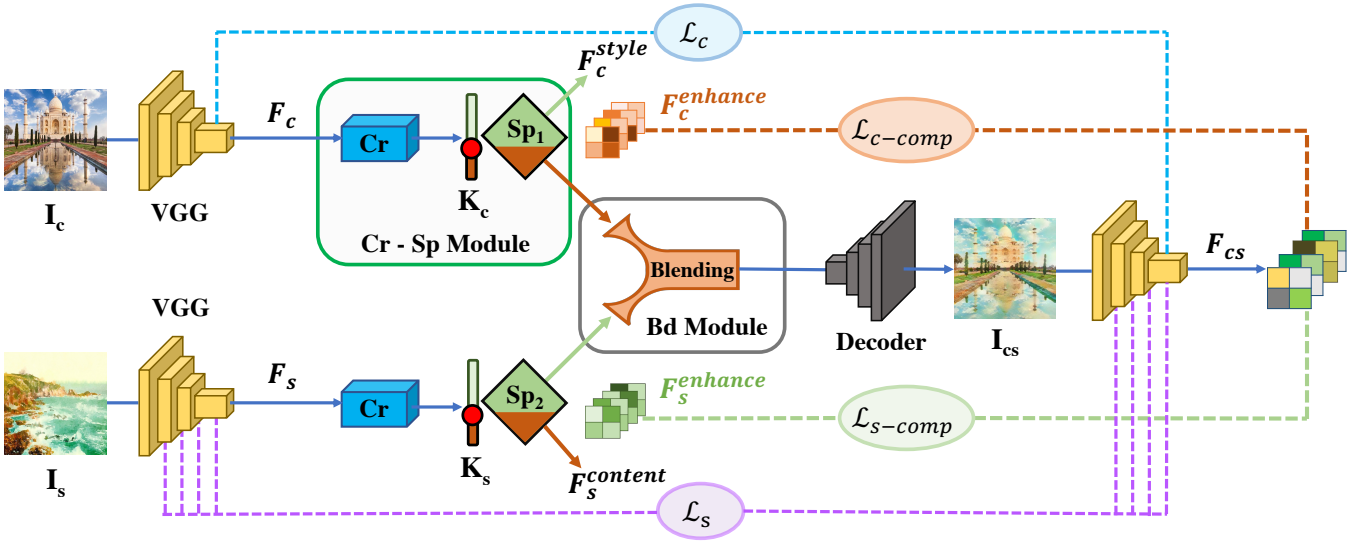


Figure 1: Overview framework of CSBNet. Cr: Crystallization; Sp: Separation; Bd: Blending.

In the following step, the style component F_c^{style} of F_c and the content component $F_s^{content}$ of F_s will be removed, and we only use the content component $F_c^{content}$ of F_c and style component F_s^{style} of F_s to generate the stability-enhanced content feature $F_c^{enhance}$ and style feature $F_s^{enhance}$. They are cross-blended via the **Blending (Bd) Module** (Sec. 3.3):

$$F_{cs} = \mathrm{Bd}(F_c^{enhance}, F_s^{enhance}). \tag{3}$$

Finally, the blended feature F_{cs} is decoded to obtain the stylized output I_{cs}:

$$I_{cs} = \mathrm{Decoder}(F_{cs}). \tag{4}$$

3.2 Crystallization & Separation Module

After obtaining the VGG features F_c and F_s, existing neural style transfer approaches always use these raw features directly to produce the stylized feature F_{cs}. However, we argue that for raw VGG features, the content components and style components are mixed. And directly using the mixed features will easily generate unnatural and flickering effects where the style components of the content images remain, or the content components of the style patterns appear in the form of ghosting artifacts. To generate these stability-aware content and style components and complete the separation of them, we propose two key modules in our CSBNet, *i.e.*, the **Crystallization (Cr) Module** and **Separation (Sp) Module**.

The structures of Cr and Sp modules are shown in Fig. 3, where these two modules are closely related.

Crystallization (Cr) Module. Given a raw VGG feature $F \in \{F_c, F_s\} \in \mathbb{R}^{C\times H\times W}$ (where C is the number of channels, H and W are height and width, respectively), we first flatten it to $\hat{F} \in \mathbb{R}^{C\times HW}$ and center $\hat{F}$ by subtracting its channel-wise mean.

Then we perform orthogonal projection on $\hat{F}$ via singular value decomposition (SVD):

$$SVD(\hat{F}) = U\Sigma V^T = \sum_{i=1}^{N} \sigma_i u_i v_i^T, \tag{5}$$

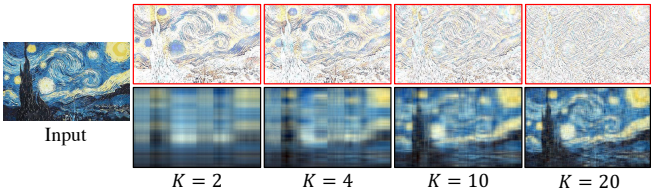


Figure 2: Visualization of content components (top) and style components (bottom) in pixel-level crystallization and separation.

where $N = min(C, HW)$, σ_i is singular value, $u_i \in \mathbb{R}^{C\times 1}$ and $v_i \in \mathbb{R}^{HW\times 1}$ are the i^{th} column of U and V, respectively. We define $A_i = \sigma_i u_i v_i^T \in \mathbb{R}^{C\times HW}$ and a matrix set $\Phi = \{A_1, A_2, ..., A_N\}$. According to the properties of SVD, for every A_i, A_j $(i \neq j)$ in Φ, we have:

$$A_i^T A_j = (\sigma_i u_i v_i^T)^T \sigma_j u_j v_j^T = \sigma_i \sigma_j (v_i u_i^T u_j v_j^T) = 0, \tag{6}$$

which means A_i and A_j are orthogonal. In experiments, we find these orthogonal matrices can represent different content and style components hierarchically (see later "Separation (Sp) Module"). These matrices are stability-aware, *e.g.*, if we separate and remove the style component of the content feature and the content component of the style feature, we can fundamentally enhance the transfer stability. Therefore, we design the Sp Module for this purpose. For easy identification, we call these orthogonal matrices crystal nuclei.

Separation (Sp) Module. This module is designed to separate the above orthogonal crystal nuclei into the content component $F^{content}$ and the style component F^{style}:

$$F^{content} = \Theta \sum_{i\geq K} A_i, \qquad F^{style} = \Theta \sum_{i<K} A_i, \tag{7}$$

where K is an empirically determined cursor to decide the degree of separation between the content and style components. Θ is the unflatten operation to reshape the feature size to $C \times H \times W$.

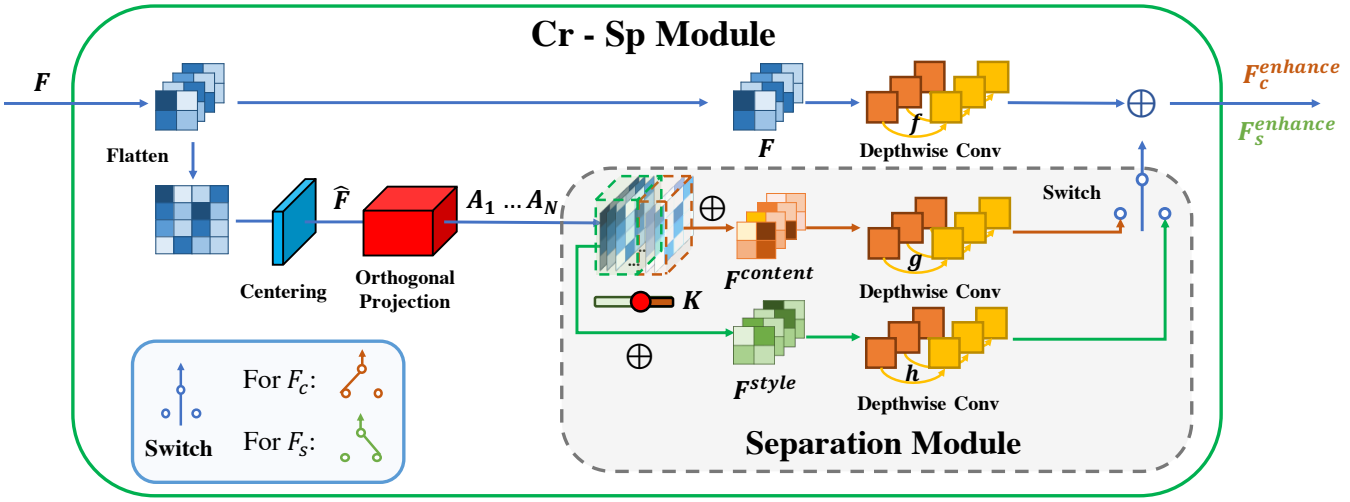


Figure 3: Structures of Crystallization (Cr) Module and Separation (Sp) Module.

To demonstrate the motivation and rationality of this separation, we visualize the pixel-level content components and style components in Fig. 2. As can be observed, with the increase of K, the content component (top row) filters out more style information and preserve more pure structural details. In contrast, the style component (bottom row) adds more detailed style information and filters out certain content information. In this way, we can further separate the content and style information to obtain more pure content and style features, which may be helpful to improve the stylization quality and transfer stability.

Specifically, we set $K = K_c$ to separate the crystal nuclei of the content feature F_c, and $K = K_s$ to separate the crystal nuclei of the style feature F_s. Thus we can easily control the content-style separation for the content image and style image via adjusting K_c and K_s, respectively.

After separation, the style component F_c^{style} of F_c and the content component $F_s^{content}$ of F_s are removed, and we only use the content component $F_c^{content}$ of F_c and the style component F_s^{style} of F_s to generate the stability-enhanced content feature $F_c^{enhance}$ and style feature $F_s^{enhance}$:

$$\begin{aligned} F_c^{enhance} &= f_1(F_c) + g(F_c^{content}), \\ F_s^{enhance} &= f_2(F_s) + h(F_s^{style}), \end{aligned} \tag{8}$$

where $f_1(\cdot)$, $f_2(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are 1×1 learnable depthwise convolution layers. We use depthwise convolution here in order to maintain the hierarchical independence of each channel. Besides, depthwise convolution also offers less computational complexity. As shown in Fig. 3, a double-pole switch is used to generate $F_c^{enhance}$ and $F_s^{enhance}$ based on whether the input feature is from the content or style image.

3.3 Blending Module

We further design a novel **Blending (Bd) Module** to blend the stability-enhanced content feature $F_c^{enhance}$ with the style feature $F_s^{enhance}$ more effectively. Our Bd module is based on MCCNet [Deng *et al.*, 2021], which considers the multi-channel correlation to achieve more stable feature alignment:

$$f_{cs}^i = (1 + \sum_{k=1}^{C} w_k ||f_s^k||_2) f_c^i, \tag{9}$$

where $f^{i/k} \in \mathbb{R}^{1\times HW}$ is the i^{th}/k^{th} channel of flattened feature $F \in \{\hat{F}_c, \hat{F}_s, \hat{F}_{cs}\}$, C is the number of channels and w_k represents the weight of the k-th style channel learned by a fully connected layer. While it has been verified that this operation can improve the stability and temporal consistency of video style transfer, we found it is easy to bring bar-like artifacts to the generated outputs (*e.g.*, the 1st and 5th rows in Fig. 5). It is because the MCC in Eq. (9) is sub-optimal, which only learns partial information of the Gram matrix.

For explanation, let's start with the Gram matrix of F_s:

$$\begin{aligned} Gram(F_s) &= \hat{F}_s \hat{F}_s^T \\ &= \begin{pmatrix} f_s^1(f_s^1)^T & f_s^1(f_s^2)^T & \cdots & f_s^1(f_s^C)^T \\ f_s^2(f_s^1)^T & f_s^2(f_s^2)^T & \cdots & f_s^2(f_s^C)^T \\ \vdots & \vdots & \ddots & \vdots \\ f_s^C(f_s^1)^T & f_s^C(f_s^2)^T & \cdots & f_s^C(f_s^C)^T \end{pmatrix} \end{aligned} \tag{10}$$

It can be easily derived that the weighted item $||f_s^k||_2$ in Eq. (9) equals $f_s^k(f_s^k)^T$, which is the k^{th} diagonal element of the Gram matrix in Eq. (10). This indicates that MCCNet conducts style transfer based on only C diagonal elements of Gram matrix, while $C \times C - C$ non-diagonal elements, which also represent the correlation between channels, are abandoned. Thus it may lead to degraded transfer results.

Our proposed Bd Module overcomes this problem through better capturing the associations between different style channels. As shown in Fig. 4, we first normalize and tansform the enhanced content feature $F_c^{enhance}$ and the style feature $F_s^{enhance}$ as follows:

$$\begin{aligned} \widetilde{F_c} &= \mathcal{J}(Norm(F_c^{enhance})), \\ \widetilde{F_s} &= \mathcal{K}(Norm(F_s^{enhance})), \end{aligned} \tag{11}$$

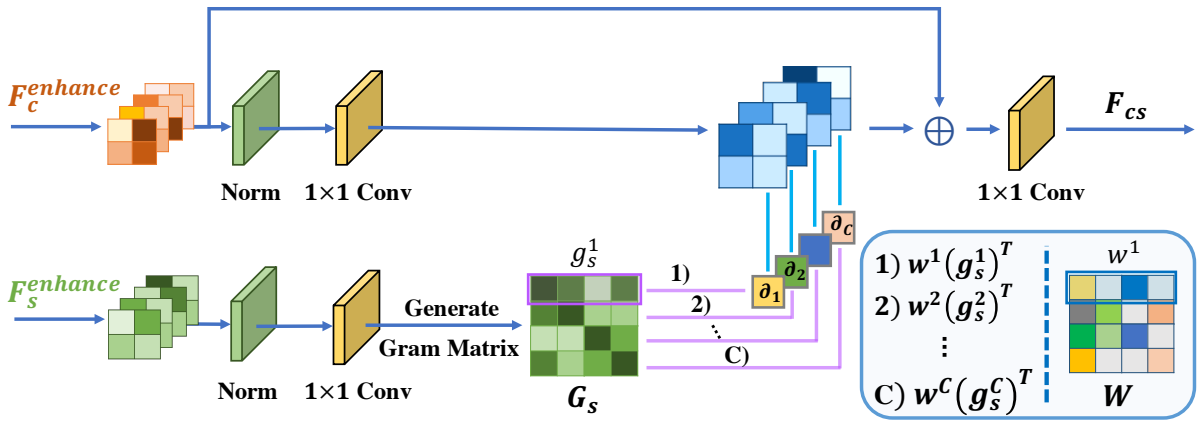


Figure 4: Structure of our Blending (Bd) Module.

where $\mathcal{J}(\cdot)$ and $\mathcal{K}(\cdot)$ are 1×1 learnable convolution layers, and $Norm(\cdot)$ is channel-wise mean-variance normalization.

Then for $\widetilde{F_s}$, we calculate its Gram matrix $G_s = Gram(\widetilde{F_s})$ and weight each row g_s^i of G_s:

$$\partial_i = w^i (g_s^i)^T, \quad i = 1, 2, ..., C, \tag{12}$$

where $w^i \in \mathbb{R}^{1\times C}$ is the i^{th} row of $W \in \mathbb{R}^{C\times C}$, a parameter-learnable matrix. Based on this, we further obtain the fusion component $\widetilde{F_{cs}}$. The i^{th} column of $\widetilde{F_{cs}}$ is as follows:

$$\widetilde{f_{cs}^i} = \partial_i \widetilde{f_c^i}, \tag{13}$$

where $\widetilde{f^i} \in \mathbb{R}^{1\times H\times W}$ is the i^{th} channel of $\widetilde{F} \in \{\widetilde{F_{cs}}, \widetilde{F_c}\}$. The final fusion feature F_{cs} can be obtained as:

$$F_{cs} = \mathcal{R}(F_c^{enhance} + \widetilde{F_{cs}}), \tag{14}$$

where $\mathcal{R}(\cdot)$ is a 1×1 learnable convolution layer.

3.4 Loss Function

We train our CSBNet using the following total loss function consisting of a peceptual loss $\mathcal{L}_{percp}$, our proposed component enhancement loss $\mathcal{L}_{comp}$, and a smooth loss $\mathcal{L}_{smooth}$:

$$\mathcal{L}_{total} = \mathcal{L}_{percp} + \mathcal{L}_{comp} + \mathcal{L}_{smooth}, \tag{15}$$

Perceptual Loss. To achieve style transfer, we use a pre-trained VGG-19 to extract content and style features and compute the content and style perceptual losses similar to AdaIN [Huang and Belongie, 2017]:

$$\mathcal{L}_{percep} = \lambda_{content}\mathcal{L}_{content} + \lambda_{style}\mathcal{L}_{style}, \tag{16}$$

where $\lambda_{content}$ and λ_{style} are hyper-parameters to balance the content and style performance.

Specifically, the content perceptual loss is used to minimize the content differences between the generated images I_{cs} and content images I_c:

$$\mathcal{L}_{content} = \sum_i ||\phi_i(I_{cs}) - \phi_i(I_c)||_2, \tag{17}$$

where ϕ_i represents the i^{th} layer in VGG-19, here we use $Relu4\_1$ and $Relu5\_1$ to calculate $\mathcal{L}_{content}$.

The style perceptual loss minimizes the style differences between the generated images I_{cs} and style images I_s:

$$\begin{aligned}\mathcal{L}_{style} = &\sum_i ||\mu(\phi_i(I_{cs})) - \mu(\phi_i(I_s))||_2 \\ &+ \sum_i ||\sigma(\phi_i(I_{cs})) - \sigma(\phi_i(I_s))||_2,\end{aligned} \tag{18}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the channel-wise mean and standard deviation, respectively. And we use $Relu1\_1$, $Relu2\_1$, $Relu3\_1$, and $Relu4\_1$ to calculate $\mathcal{L}_{style}$.

Component Enhancement Loss. In order to enhance the quality of content-style component separation in the network, we design a novel pair of component enhancement losses, consisting of a content componet loss $\mathcal{L}_{c-comp}$ and a style component loss $\mathcal{L}_{s-comp}$:

$$\mathcal{L}_{comp} = \lambda_{c-comp}\mathcal{L}_{c-comp} + \lambda_{s-comp}\mathcal{L}_{s-comp}, \tag{19}$$

where λ_{c-comp} and λ_{s-comp} are hyper-parameters.

We hope to minimize the content and style differences of corresponding components. Thus, we define the content component loss as the content differences between the stability-enhanced content features $F_c^{enhance}$ and the generated features F_{cs}:

$$\mathcal{L}_{c-comp} = ||F_c^{enhance} - F_{cs}||_2. \tag{20}$$

Similarly, the style component loss is used to minimize the style differences between the stability-enhanced style features $F_s^{enhance}$ and the generated features F_{cs}:

$$\begin{aligned}\mathcal{L}_{s-comp} = &||\mu(F_s^{enhance}) - \mu(F_{cs})||_2 \\ &+ ||\sigma(F_s^{enhance}) - \sigma(F_{cs})||_2.\end{aligned} \tag{21}$$

With these component enhancement losses, our network can better learn the stability-enhanced content and style information, thus effectively suppressing the generation of artifacts and improving the stylization quality as well as stability (see ablation studies in later Sec. 4.4).

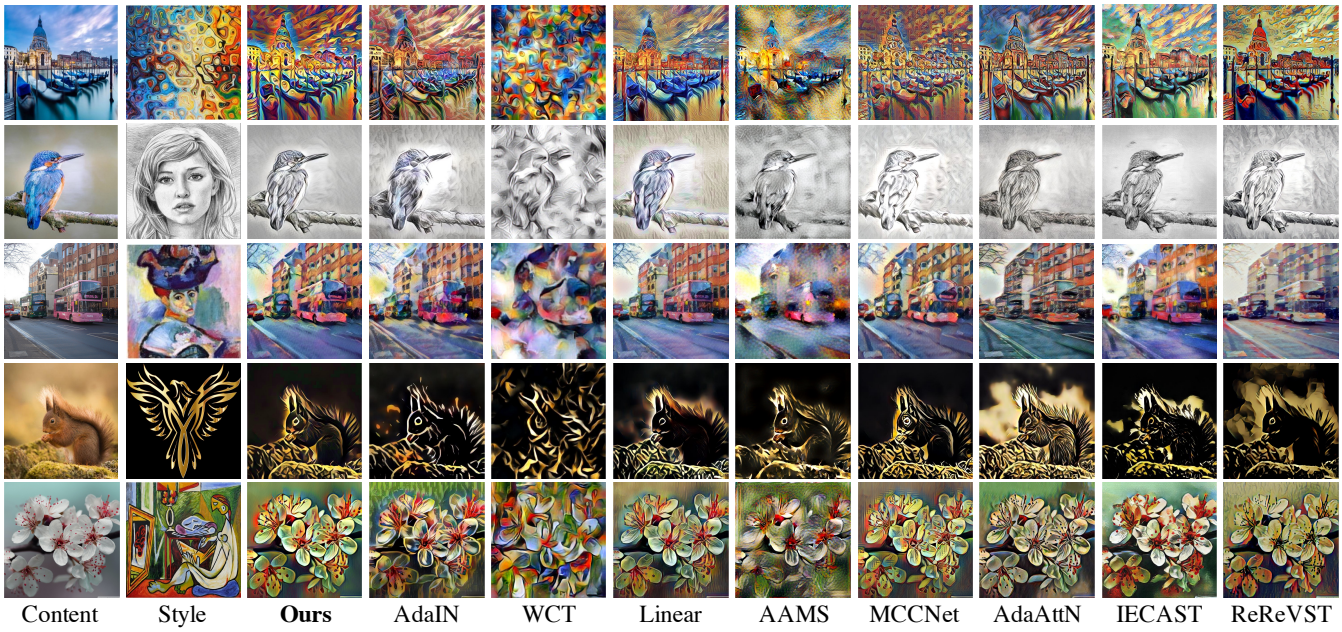


Figure 5: Qualitative comparisons of universal image and video style transfer methods.

Smooth Loss. Similar to MCCNet [Deng *et al.*, 2021], we also adopt a total variation loss $\mathcal{L}_{tv}$ [Johnson *et al.*, 2016] and an illumination loss $\mathcal{L}_{illum}$ to smooth the results and improve the robustness to complex light conditions in input videos.

$$\mathcal{L}_{smooth} = \lambda_{tv}\mathcal{L}_{tv} + \lambda_{illum}\mathcal{L}_{illum}, \tag{22}$$

where λ_{tv} and λ_{illum} are hyper-parameters. The illumination loss $\mathcal{L}_{illum}$ is defined as:

$$\mathcal{L}_{illum} = ||I_{cs}^{noise} - I_{cs}||_2, \tag{23}$$

where I_{cs}^{noise} is the results generated with noisy content image $I_c + \theta$ (θ is random Gaussian noise) and style image I_s.

4 Experimental Results

4.1 Implementation Details

The hyper-parameters in our model are set to $\lambda_{content} = 3$, $\lambda_{style} = 10$, $\lambda_{c-comp} = 3$, $\lambda_{s-comp} = 1$, $\lambda_{tv} = 0.0001$, and $\lambda_{illum} = 3000$. The separation cursors are empirically set to $K_c = 4$, $K_s = -10$ (here '-' represents the inverted index of array). We choose MS-COCO [Lin *et al.*, 2014] as content image set and WikiArt [Phillips and Mackintosh, 2011] as style image set. During the training stage, the input images are resized to 512×512 pixels and randomly cropped to 256×256 pixels. Batch size is chosen as 4 for 160,000 iterations.

4.2 Qualitative Comparisons

Eight SOTA universal image and video style transfer methods are selected for comparison, including AdaIN [Huang and Belongie, 2017], WCT [Li *et al.*, 2017], Linear [Li *et al.*, 2019], AAMS [Yao *et al.*, 2019], MCCNet [Deng *et al.*, 2021], AdaAttN [Liu *et al.*, 2021], IECAST [Chen *et al.*, 2021a], and ReReVST [Wang *et al.*, 2020a].

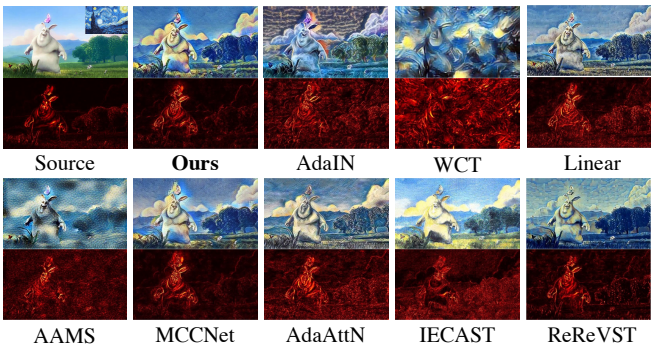


Figure 6: Qualitative comparisons of video style transfer. The first row shows the stylized video frames. The second row shows the heat maps visualizing the average differences between adjacent frames.

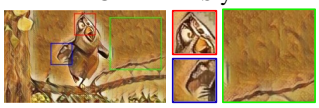
Image Style Transfer. The comparisons of image stylizations are shown in Fig. 5. The over-simplified alignment of AdaIN easily leads to style artifacts and content distortion (*e.g.* 2nd and 5th rows). Due to global parameter transformation, WCT seriously damages the content structures in the results. Linear can better preserve content affinity, but the degree of stylization is low (*e.g.* 1st and 2nd rows). AAMS, MCCNet, AdaAttN, and IECAST are essentially based on the attention mechanism. However, point-like artifacts appear in AAMS (*e.g.* 1st, 3rd, and 5th rows), and bar-like artifacts appear in MCCNet (*e.g.* 1st and 5th rows). AdaAttN preserves the content structures well but still suffers from noticeable artifacts (*e.g.* 3rd and 5th rows) and poor transfer of style color distributions (*e.g.* 1st and 3rd rows). IECAST is prone to produce eye-like artifacts (*e.g.* 2nd and 3rd rows). ReReVST is designed for video style transfer, which often produces less

Metrics	Input	**Ours**	AdaIN	WCT	Linear	AAMS	MCCNet	AdaAttN	IECAST	ReReVST
E_c	0.000	1.874	2.393	3.541	**1.856**	2.119	2.051	2.158	1.930	1.988
E_s	3.241	**1.130**	1.135	1.560	1.280	1.142	1.165	1.249	1.458	1.523
LPIPS	0.071	**0.108**	0.190	0.432	0.117	0.297	0.122	0.219	0.245	0.137
D^*	30.78	**47.50**	80.88	113.71	63.75	86.48	51.13	71.46	91.30	54.80
Time/(sec)	—	0.017	0.008	0.579	0.013	2.173	0.015	0.018	0.011	0.098

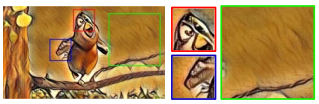
Table 1: The average metrics of inputs and stylized results of different methods on 10 video clips. The first and second rows are E_c and E_s, evaluating stylization quality. The third and fourth rows are LPIPS and D^*, evaluating video consistency. The last row is the average transfer time for a single 512×512 frame, evaluating transfer efficiency. These metrics all follow the rule: the lower, the better.

Content & Style

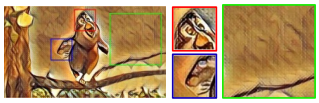
(a) Ours



(b) w/o Cr-Sp For F_c



(c) w/o Cr-Sp For F_s

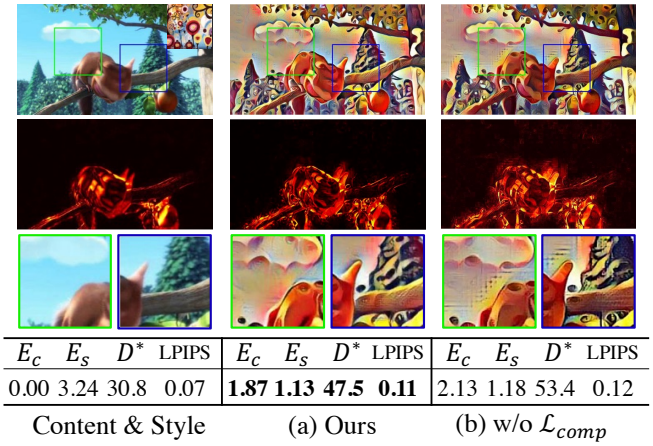


(d) w/o Cr-Sp For F_c & F_s

	E_c	E_s	D^*	LPIPS
Ours	**1.874**	**1.130**	**47.50**	**0.108**
(b)	2.353	1.137	55.38	0.129
(c)	2.171	1.234	69.12	0.111
(d)	2.323	1.159	63.19	0.118

(e) quantitative comparison

Figure 7: Ablation study of Crystallization and Separation (Cr-Sp) Module.



E_c	E_s	D^*	LPIPS	E_c	E_s	D^*	LPIPS	E_c	E_s	D^*	LPIPS
0.00	3.24	30.8	0.07	**1.87**	**1.13**	**47.5**	**0.11**	2.13	1.18	53.4	0.12

Content & Style (a) Ours (b) w/o $\mathcal{L}_{comp}$

Figure 8: Ablation study of component enhancement loss.

stylized results for image stylizations.

In contrast to these SOTA approaches, the results of our CSBNet exhibit higher stylization quality. The active separation of content and style components by the Cr and Sp Modules successfully suppresses the appearance of style artifacts and better maintains the content structures. Moreover, the better utilization of style information by the Bd Module leads to a higher degree of stylization in the results.

Video Style Transfer. The comparisons of video stylizations are shown in Fig. 6. We present heatmaps to show the average differences between adjacent frames. It can be seen that the difference of our results is closest to that of the input frames, obviously outperforming other methods in terms of temporal stability and consistency. This is due to the fact that our CSBNet can effectively maintain the content structures of the video frames and suppress the generation of random artifacts causing frame-flicker.

4.3 Quantitative Comparisons

Stylization Quality. We leverage the content and style errors E_c and E_s of [Gatys *et al.*, 2016] to quantitatively evaluate the stylization quality. As shown in the top two rows of Tab. 1, CSBNet enjoys the lowest style error E_s among 9 methods, while content error E_c is only slightly higher than Linear [Li *et al.*, 2019], ranking the second. Metrics E_c and E_s show that our method can effectively improve stylization quality while better preserving content structures, leading to comprehensively more satisfying transfer results.

Video Consistency. We adopt LPIPS (*Learned Perceptual Image Patch Similarity*) [Zhang *et al.*, 2018] and D^* [Deng *et al.*, 2021] to quantify the stability and consistency of video style transfer. Given a video clip with N frames, we compute the average perceptual distances between adjacent frames via LPIPS, and the average pixel distances via D^*, *i.e.*, $D^* = \frac{1}{N-1}\sum_{t=0}^{N-2} ||F_{t+1} - F_t||_2$. The results shown in the middle two rows of Tab. 1 further reflect the superiority of our CSBNet on video style transfer, which is consistent with the qualitative results in Fig. 6.

Efficiency. As shown in the last row of Tab. 1, CSBNet achieves comparable speed with SOTA methods, only slightly behind AdaIN, Linear, MCCNet, and IECAST. However, we believe it is worthwhile to sacrifice slight run time speed for higher stylization quality and improvement of consistency.

4.4 Ablation Studies

Crystallization and Separation (Cr-Sp) Module. Fig. 7 shows the results of video style transfer with and without Cr-Sp module for F_c and F_s, respectively. It can be found that (b) without Cr-Sp for F_c, the unstable strip and dot artifacts are generated in the results; (c) without Cr-Sp for F_s, there will be a loss of content details (*e.g.*, the feather and eyes of the bird) and the blue color of the content image leak into the result (*e.g.*, the eyes of the bird). In contrast, with Cr-Sp for both F_c and F_s, our method generates smoother and more satisfying results with better-preserved content structures and

K_c	E_c	E_s	D^*	LPIPS
0	2.07	1.27	52.75	0.116
2	1.94	1.61	48.92	0.111
4	**1.87**	**1.13**	**47.50**	**0.108**
10	1.90	1.16	49.78	0.112
15	1.91	1.21	51.39	0.116
20	1.92	1.33	51.30	0.118
200	2.08	1.32	51.79	0.119

(a) Effects of K_c ($K_s = -10$)

K_s	E_c	E_s	D^*	LPIPS
-1	1.94	1.23	48.45	0.115
-4	1.95	1.28	48.40	0.113
-10	**1.87**	**1.13**	**47.50**	**0.108**
-15	1.95	1.15	49.47	0.110
-20	1.99	1.15	53.80	0.115
-30	2.07	1.14	54.50	0.118
-200	2.11	1.18	54.00	0.117

(b) Effects of K_s ($K_c = 4$)

Table 2: Effects of different settings of K_c and K_s.

fewer local artifacts. This observation is consistent with the quantitative scores shown in column (e), which validates that our Cr-Sp module indeed helps improve the stylization quality and temporal stability.

Component Enhancement Loss. A pair of new losses is introduced in our framework for enhancing transfer results, and thus we further conduct ablation experiments on it. From Fig. 8, we can observe that without the proposed component enhancement loss, the edge of the generated result is prone to produce grid-like artifacts (*e.g.*, the edges of tree and the squirrel's ear and back), which not only degrades the stylization quality, but also reduces the transfer stability. Best viewed in conjunction with the quantitative scores below.

Setting of K_c and K_s. We also conduct ablation experiments to demonstrate the effects of different settings of our separation cursors K_c and K_s. We can observe from Tab. 2 that increasing the values of $|K_c|$ and $|K_s|$ within a certain range not only helps to reduce the content and style errors E_c and E_s, but also increases the temporal stability of output videos (lower D^* and LPIPS scores). It is because increasing the value of $|K_c|$ can filter out the style components of content images, thus helping improve the stylization of the results. Meanwhile, increasing the value of $|K_s|$ can filter out the content components of style images, thus helping preserve the integrity of the content structures and improving the videos' fluency. However, too high values of $|K_c|$ and $|K_s|$ will reduce the stylized quality and stability of the output videos.

5 Conclusion

In this work, we propose a novel frame-based network, termed CSBNet, for universal video style transfer. Three innovative modules, *i.e.*, the Crystallization (Cr) Module, Separation (Sp) Module, and Blending (Bd) Module, and a new pair of component enhancement losses are introduced to improve video consistency and achieve higher-quality stylization. Extensive qualitative and quantitative experiments are conducted to demonstrate the effectiveness and superiority of our CSBNet against state-of-the-art algorithms.

References

[Chen *et al.*, 2017] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent online video style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1105–1114, 2017.

[Chen *et al.*, 2021a] Haibo Chen, Lei Zhao, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. Artistic style transfer with internal-external learning and contrastive learning. In *Advances in Neural Information Processing Systems*, 2021.

[Chen *et al.*, 2021b] Haibo Chen, Lei Zhao, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. Dualast: Dual style-learning networks for artistic style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 872–881, 2021.

[Deng *et al.*, 2021] Yingying Deng, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, and Changsheng Xu. Arbitrary video style transfer via multi-channel correlation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1210–1217, 2021.

[Gao *et al.*, 2020] Wei Gao, Yijun Li, Yihang Yin, and Ming-Hsuan Yang. Fast video multi-style transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3222–3230, 2020.

[Gatys *et al.*, 2016] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[Huang and Belongie, 2017] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[Johnson *et al.*, 2016] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[Li and Wand, 2016] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2479–2486, 2016.

[Li *et al.*, 2017] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30, 2017.

[Li *et al.*, 2019] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3809–3817, 2019.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[Liu *et al.*, 2021] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li,

and Errui Ding. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6649–6658, 2021.

[Park and Lee, 2019] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5880–5888, 2019.

[Phillips and Mackintosh, 2011] Fred Phillips and Brandy Mackintosh. Wiki art gallery, inc.: A case for critical thinking. *Issues in Accounting Education*, 26(3):593–608, 2011.

[Ruder *et al.*, 2016] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *German conference on pattern recognition*, pages 26–36. Springer, 2016.

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

[Ulyanov *et al.*, 2017] Dmitry Ulyanov, Andrea Vedaldi, and Victor 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*, pages 6924–6932, 2017.

[Wang *et al.*, 2020a] Wenjing Wang, Shuai Yang, Jizheng Xu, and Jiaying Liu. Consistent video style transfer via relaxation and regularization. *IEEE Transactions on Image Processing*, 29:9125–9139, 2020.

[Wang *et al.*, 2020b] Zhizhong Wang, Lei Zhao, Haibo Chen, Lihong Qiu, Qihang Mo, Sihuan Lin, Wei Xing, and Dongming Lu. Diversified arbitrary style transfer via deep feature perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7789–7798, 2020.

[Wang *et al.*, 2021] Zhizhong Wang, Lei Zhao, Haibo Chen, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. Evaluate and improve the quality of neural style transfer. *Computer Vision and Image Understanding*, 207:103203, 2021.

[Yao *et al.*, 2019] Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, and Jun Wang. Attention-aware multi-stroke style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1467–1475, 2019.

[Zhang *et al.*, 2018] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.