Consistent Arbitrary Style Transfer Using Consistency Training and Self-Attention Module

Zheng Zhou, Yue Wu, *Member, IEEE*, and Yicong Zhou, *Senior Member, IEEE*

***Abstract*—Arbitrary style transfer (AST) has garnered considerable attention for its ability to transfer styles infinitely. Although existing methods have achieved impressive results, they may overlook style consistencies and fail to capture crucial style patterns, leading to inconsistent style transfer (ST) caused by minor disturbances. To tackle this issue, we conduct a mathematical analysis of inconsistent ST and develop a style inconsistency measure (SIM) to quantify the inconsistencies between generated images. Moreover, we propose a consistent AST (CAST) framework that effectively captures and transfers essential style features into content images. The proposed CAST framework incorporates an intersection-of-union-preserving crop (IoUPC) module to obtain style pairs with minor disturbance, a self-attention (SA) module to learn the crucial style features, and a style inconsistency loss regularization (SILR) to facilitate consistent feature learning for consistent stylization. Our proposed framework not only provides an optimal solution for consistent ST but also outperforms existing methods when embedded into the CAST framework. Extensive experiments demonstrate that the proposed CAST framework can effectively transfer style patterns while preserving consistency and achieve the state-of-the-art performance.**

***Index Terms*—Arbitrary style transfer (AST), consistent training, self-attention (SA), style inconsistency.**

I. Introduction

Style transfer aims to build a new image which migrates the style patterns in a style image onto the contents of another image [1]. Recently, the neural network methods achieved great success in image processing [2], [3], [4], [5], [6]. Many researchers devoted to developing effective neural style transfer (ST) models to render a generated image with different styles using the convolutional neural networks [7]. Gatys et al. [8] proposed the neural ST method using the pretrained visual geometry group (VGG) [9] model to extract the deep features and represented the style features (e.g., color and drawing) using the Gram matrix. However, this method simply combines the content and style together using iterative optimization.

With the rapid development of deep neural networks, neural ST models can stylize images offline using the trained deep learning models. According to the ST capacity, neural ST models are divided into three categories: 1) *a certain style per model:* Johnson et al. [10] introduced a feedforward convolutional neural network to transfer style features into the content image with a perceptual loss and trained one transformation model for each style image; 2) *multiple styles per model*: Chen et al. [11] developed a style bank module to train the convolution filters for many kinds of style and transfer the pretrained style into the content image; and 3) *arbitrary style per model*: Huang and Belongie [12] proposed the adaptive instance normalization (AdaIN) layer to shift the feature statistics from style to content, and trained one model for any style transferring. One successful transformation model should transfer style as much as possible. Apparently, the most attractive one is the arbitrary style per model due to its high efficiency and capability of transferring any style using one trained model [13]. Therefore, the arbitrary ST (AST) has achieved a flourish development. Jing et al. developed dynamic instance normalization (DIN) [14] to learn the flexible convolution kernel and bias parameters from style in a more sophisticated way for ST. Li et al. [15] integrated the whiten and color transforms (WCT) into the reconstruction model to synthesize satisfactory stylized results. Linear ST (LST) [16] was proposed to learn the transformations in a symmetric autoencoder module to address the issue of complicated matrix manipulation in WCT [15]. Park and Lee [17] developed style-attentional network (SAN) to capture richer style patterns. Deng et al. [18] introduced the self-adaptation module into SAN to propose a multi-adaptation network (MAN). Liu et al. [19] proposed the adaptive attention normalization (AdaAttN) to align the point-wise feature statistics. Singh et al. [20] combined the self-attentive factorized instance normalization (SAFIN) to remove the unwanted artifacts in the generated images.

The above-mentioned methods have achieved successful ST results with abundant style features. However, these models are still plagued by the inconsistent ST, leading to unstable style migration and unsatisfactory performance [21]. To address these issues, many consistent models were developed for ST. For instance, Wang et al. [21] proposed a relaxation loss and a regularization strategy for consistent ST. However, existing consistent ST models mainly focus on dealing with temporal inconsistencies that are inapplicable for spatial inconsistency caused by the style disturbance. As shown in Fig. 1, the style spatial inconsistency is mainly presented in color mixing

Manuscript received 7 March 2022; revised 2 April 2023; accepted 18 July 2023. Date of publication 4 August 2023; date of current version 30 October 2024. This work was supported by the Science and Technology Development Fund, Macau SAR (File no. 0049/2022/A1), and by the University of Macau (File no. MYRG2022-00072-FST). *(Corresponding author: Yicong Zhou.)*

Zheng Zhou and Yicong Zhou are with the Department of Computer and Information Science, University of Macau, Macau, China (e-mail: alexzhouzhengz@gmail.com; yicongzhou@um.edu.mo).

Yue Wu is with the Amazon Alexa Natural Understanding, Manhattan Beach, CA 90007 USA (e-mail: wuayue@amazon.com).

Digital Object Identifier 10.1109/TNNLS.2023.3298383

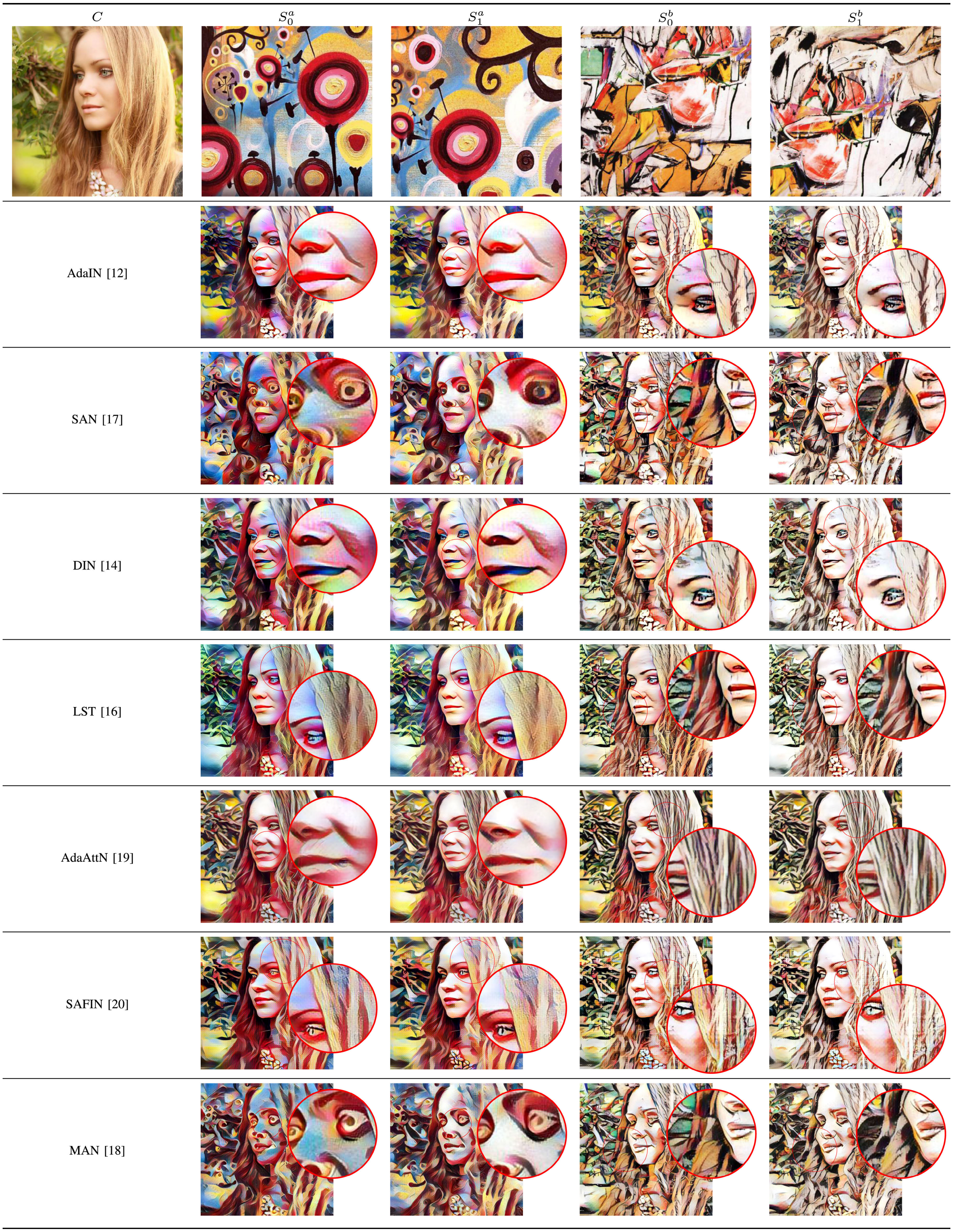

Fig. 1. Inconsistent stylization results of AST models. The first row shows the content image c and style images denoted by S_0^i and S_1^i, $i \in a, b$. The second to eighth rows show the stylization results of AdaIN [12], SAN [17], DIN [14], LST [16], AdaAttN [19], SAFIN [20], and MAN [18], respectively. Small inconsistency between similar style images results large disturbances between the generated images under style images S_0^i and S_1^i, $i \in a, b$.

and shape deformation. For example, the stylization results of AdaIN [12] show the unwanted style artifacts near the human nose and hair areas. Visual results of SAN [17] contain the color differences and content distortions in the human face. LST [16] performs the slight color depth inconsistency in the forehead areas. These problems are mainly caused by style spatial inconsistency that leads to the failure of capturing the core style patterns (e.g., color and texture) and makes models sensitive to style disturbance. Therefore, in addition to the temporal consistent ST, a spatial consistent ST is also important to improve the model performance. Previous studies have mainly focused on effectively synthesizing spatial style features onto content images [22], [23], [24]. Chen et al. [22] proposed a feed-forward network that consistently transfers style to different content image views for stereoscopic neural ST. Yao et al. [23] introduced an attention network that integrates multiple stroke patterns into different spatial regions of the content image. Chang and Chen [24] developed a relation loss and a spatial relation module to capture long-range dependency and reduce artifacts. However, these approaches do not explicitly consider the issue of inconsistent stylization caused by minor variations in the style image. In this work, we investigate the impact of minor disturbance on style spatial consistency in ST and introduce a consistent AST (CAST) framework to preserve style consistency and achieve satisfactory stylization results. To the best of our knowledge, our work is the first to tackle this specific problem in the field of consistent ST. The main contributions of this work are summarized as follows.

1) We mathematically analyze the problem of style inconsistency caused by style disturbance and develop a style inconsistency measure (SIM) to quantitatively evaluate the degree of style inconsistency.
2) Based on this SIM, we propose a CAST framework to effectively transfer essential style patterns into content images. CAST introduces an intersection over union preserving crop module to obtain style image pairs with minor disturbance, a self-attention (SA) module to capture the essential features, and a style inconsistency loss to regularize the learning process of consistent features for CAST.
3) The proposed CAST framework not only provides an optimal solution for consistent ST but also obtains significantly better performance than existing ST methods when they are embedded into our CAST framework.
4) Extensive experiments demonstrate that the proposed CAST framework outperforms the state-of-the-art methods in transferring style patterns while preserving consistency.

The rest of this article is organized as follows. Section II reviews the related works of the ST and SA module. In Section III, we analyze the style inconsistency problem and propose the SIM. Section IV introduces the consistent ST framework. Qualitative and quantitative experiment results are presented in Section V and ablation study of the proposed model is given in Section VI. Finally, Section VII concludes this article.

II. Related Works

In this section, we briefly review the related work of ST and SA module.

A. Style Transfer

Given the content and style images, the primary goal of ST is to transfer the style features onto the content images. A good ST model is to achieve the balance between the content and style simultaneously [17]. Recently, researchers have been devoted into developing AST models to improve efficiency and performance of the ST using deep neural networks [25]. For example, Huang and Belongie [12] proposed the AdaIN which is one of the classic AST method transferring the statistical distribution of deep features from style into content in a holistic way. AdaIN adopts the following style and content loss:

$$\begin{aligned}\mathcal{L}^{l}_{\text{style}}(G,S) &= \|\mu(\phi_l(G))-\mu(\phi_l(S))\|_2\\ &\quad+\|\sigma(\phi_l(G))-\sigma(\phi_l(S))\|_2 \qquad (1)\\ \mathcal{L}^{l}_{\text{content}}(G,C) &= \|\phi_l(G)-\phi_l(C)\|_2 \qquad (2)\end{aligned}$$

where the style loss $\mathcal{L}^{l}_{\text{style}}$ in lth layer is calculated by the mean square error (MSE) of mean μ and standard deviation σ between generated image G and style image S, ϕ_l denotes the deep features in the lth layer; the content loss $\mathcal{L}_{\text{content}}$ compares the MSE between lth layer deep features (ϕ_l) of generated and content image. AdaIN adopts several rectified linear unit (ReLU) layers, including ReLU_1_1, ReLU_2_1, ReLU_3_1, ReLU_4_1 in pretrained VGG19 as style layers for the style loss, and ReLU_4_1 as content layer for the content loss. Park and Lee [17] presented an SAN that effectively captures the stylization features through the application of the non-local mechanism. Subsequently, Deng et al. [18] extended this work by introducing a multi-adaption network (MAN) that integrates channel and spatial attention into the SAN. Additionally, recent advancements in AdaAttN have been proposed, including the AdaAttN [19] and SAFIN [20] techniques. To achieve precise feature distribution matching in ST, Zhang et al. [26] employed exact histogram matching of image features in the deep space. In recent years, transformer-based methods have emerged as promising approaches, demonstrating superior performance in ST. For instance, Wu et al. [27] utilized a transformer-based network to composite style features into content images. To address the long-range dependencies within images, Deng et al. [28] proposed a transformer-based approach for ST. The existing ST methods can achieve good ST results. However, they still lack robustness to small style disturbances. Therefore, we will make a profound study on the inconsistent AST which is mainly caused by the loss of essential style patterns.

B. SA Module

SA has caught the great interest of researchers due to its ability to capture long-range dependence [29]. The methods in [23] and [30] use the query-key-value structure to calculate the attention map which reflects the response of self-content in different areas. The query-key-value features are learned through different convolutional blocks. The attention maps are obtained from the cross correlation [31] between the subspace query and key features. After the softmax activation, the normalized attention map is calculated as the guidance for the key features. Using the attention map, the models pay more attention on the area of interest to obtain a satisfactory result. Different from this, our module adopts the shareable feature embedding for the attention map and obtains the spatial attention across the channel-wise features.

III. Style Inconsistency Measure

To study the style inconsistency problem, we first systematically analyze the style spatial inconsistency and propose a SIM to quantitatively evaluate the style inconsistency of a ST model.

A good ST model can generate robust style patterns while preserving the content structure on stylized images. However, as shown in Fig. 1, the style inconsistency greatly affects the performance. Generally speaking, inconsistent ST means similar style images S_0 and S_1 with the same style patterns may produce style inconsistencies. As shown in Fig. 1, style inconsistencies between the generated images G_0 and G_1 occur and cause the color mixture, texture artifacts, and pattern deletion. Therefore, the inconsistency problem reveals that existing ST methods perform unstable ST in presence of perturbations in style images. To address this issue, we propose a SIM to describe the style inconsistencies. SIM is defined as

$$\mathrm{SIM}(\theta) = \frac{1}{\|\mathbb{S}\|}\frac{1}{K}\sum_{S\in\mathbb{S}}\sum_{k=1;S_0^k,S_1^k\in S}^{K}\frac{\nu\left(\theta\left(S_0^k\right),\theta\left(S_1^k\right)\right)}{\nu\left(S_0^k,S_1^k\right)} \tag{3}$$

where SIM denotes the evaluation function of the AST method θ which transfers a style image S within style image sets $\mathbb{S}$ into a content image. $\nu(\cdot,\cdot)$ calculates the style inconsistency between two paired images. k indicates the index of the paired images. Here, the style image $S \in \mathbb{S}$ is divided into N partitions with the same size. Every two images from these N partitions form pairs of style images S_0^k and S_1^k, and $N = 4$. Therefore, we can evaluate the consistent style transferability of AST method θ using the obtained style images S_0^k and S_1^k in the kth pair. Considering the variance between style images, we take the style inconsistency ν between S_0^k and S_1^k as the denominator which works as the balanced boundary term. For a consistent ST model, the smaller the inconsistency between style images is, the more consistent between generated images will be. When the denominator is fixed, the SIM value will increase as the inconsistency between the generated images becomes larger, and vice versa. For AST model θ, a larger SIM value always means higher style inconsistency.

In this article, function ν in (3) is calculated by the style loss function $\mathcal{L}_{\text{style}}$ in AST models

$$\nu(S_0,S_1) = \sum_{l=1}^{L}\mathcal{L}_{\text{style}}^{l}(S_0,S_1) \tag{4}$$

where $\mathcal{L}_{\text{style}}^{l}(S_0,S_1)$ indicates the style loss function between style images S_0 and S_1 at the lth feature layer. We adopt the style loss function in (1) to calculate the style inconsistency between images. Fig. 2 presents the distributions of SIM values for several popular AST methods such as AdaIN [12], SAN [17], and MAN [18]. We can observe that the quantitative evaluation SIM results of style inconsistency are consistent with the visual quality of generated images. This means that these existing AST models lack the ability of consistently migrating the style into the content image. Their SIM values are often very high. A robust AST model should have a small SIM value located in the left area of the distribution. In addition, we also test the Gram-matrix-based style loss function [8] as the style inconsistency evaluation function ν but the results show that the style loss in (1) seems more reasonable. Next, we develop a consistent ST framework to migrate important style patterns into content images robustly.

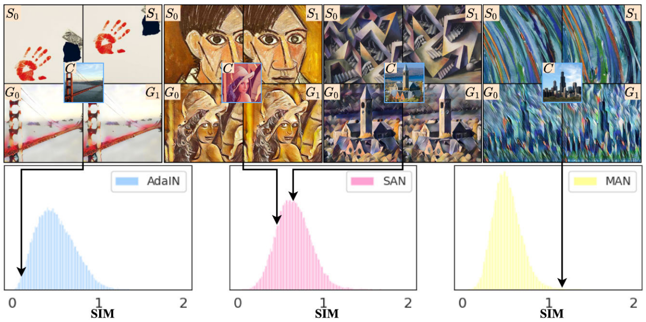

Fig. 2. Distributions of SIM for AST methods AdaIN [12], SAN [17], and MAN [18]. In each pair, C denotes the content image, S_0 and S_1 denote the style images, and G_0 and G_1 denote the corresponding generated images. The inconsistency of ST models is evaluated by SIM of the generated and style images. A larger value of SIM means the higher inconsistency of the AST method.

IV. Consistent Arbitrary Style Transfer

In this section, we propose a CAST framework to robustly transfer the important style patterns into content images. We first present the overall structure of CAST framework and then introduce our three important components: 1) an intersection-of-union (IoU) preserving random crop module to obtain the style images with the same style patterns; 2) a style inconsistency loss for style consistency training (SCT); and 3) a plug-and-play SA module to capture the important style patterns represented by SA statistics.

A. CAST Framework

The flowchart of proposed CAST framework is shown in Fig. 3. The CAST framework first inputs the style image S into the IoU-preserving random crop module to obtain two paired style samples S_0 and S_1. These paired style images S_0, S_1 and content image C are then fed into a style encoder block like VGG [12] and MoblieNet [14] to obtain deep features F_{S_0}, F_{S_1}, and F_C. A feature transformation block migrates the paired style features to content features to generate the paired stylized content features F_{CS_0} and F_{CS_1} for the decoder block to obtain the generated images G_0 and G_1. Meanwhile, a SA module is plugged into the transformation block to extract the important style patterns. The training procedure of CAST adopts the content loss and style loss in [12]. The CAST framework further proposes a new style inconsistency loss $\mathcal{L}_{\text{SI}}$ to obtain consistent stylization results. Next, Sections IV-B–IV-D present our proposed IoU-preserving random crop module, style inconsistency loss, and SA module in the CAST framework, respectively.

B. IoU-Preserving Random Crop Module

As shown in Fig. 4, the CAST framework introduces an IoU-preserving random crop module to obtain the slightly different style images S_0 and S_1 with the same style pattern for consist style training. Different from the traditional data

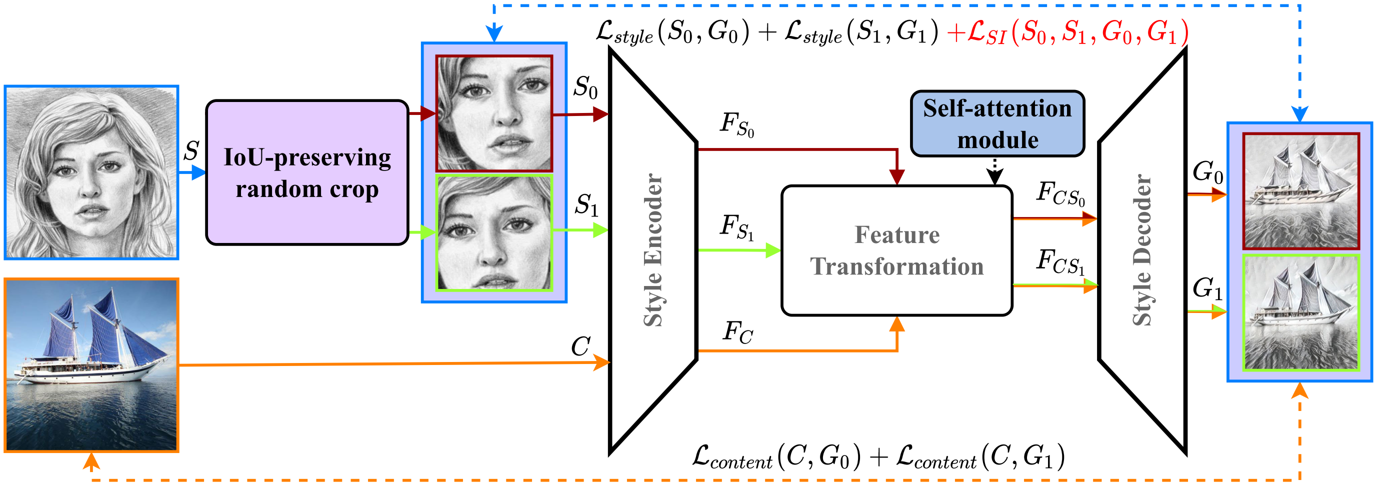
$\mathcal{L}_{style}(S_0, G_0) + \mathcal{L}_{style}(S_1, G_1) + \mathcal{L}_{SI}(S_0, S_1, G_0, G_1)$
S
IoU-preserving random crop
S_0
S_1
C
Style Encoder
F_{S_0}
F_{S_1}
F_C
Self-attention module
Feature Transformation
F_{CS_0}
F_{CS_1}
Style Decoder
G_0
G_1
$\mathcal{L}_{content}(C, G_0) + \mathcal{L}_{content}(C, G_1)$

Fig. 3. Flowchart of the proposed CAST framework. IoU-preserving random crop module provides abundant style samples. $\mathcal{L}_{\text{SI}}$ optimizes the consistency between G_0 and G_1 with important style patterns. SA module extracts the robust statistical features.

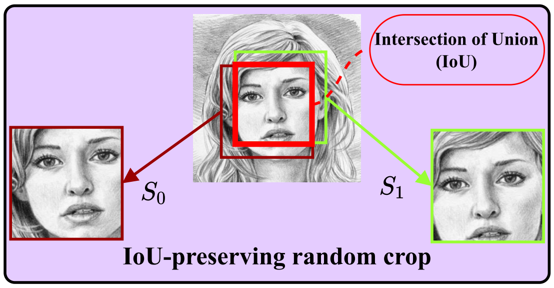

Fig. 4. IoU-preserving random crop module.

augmentation [32], our module is specially designed for ST and considers both heterogeneous and homogeneous style patterns to achieve consistent ST. As shown in Fig. 5, the overlapping region that represents the same style patterns can be evaluated by the intersection between paired style images. On the other hand, the non-overlapping region increases the style diversity between paired style images and improves the ability of consistency training. At the same time, random crop enriches the style sets to ensure that the model can achieve satisfactory results with a limited size of image dataset. The proposed IoU-preserving random crop module can not only retain the same style patterns but also consider the diversified style patterns from the same image. It aims to provide abundant style images for consistent ST training. Using the obtained style image samples with identical style patterns, the consistent model can capture the key style patterns from the image.

C. Style Inconsistency Loss

To train a robust and stable AST model, we develop a style inconsistency loss $\mathcal{L}_{\text{SI}}$ based on the proposed SIM defined in (3). For style images S_0 and S_1 with the same style patterns, an unstable AST model will generate the stylized images G_0 and G_1 with a huge inconsistency. However, a CAST model can capture key style patterns and consistently migrate them into the content image C. The SCT of AST models can be regarded as minimizing the style inconsistency between generated images with the same style patterns. Therefore, we should train a CAST model to reduce the style

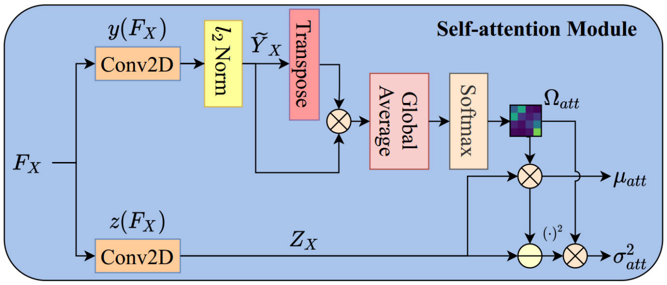

Fig. 5. Structure of the proposed plug-and-play SA module. SA statistics, mean (μ_{att}) and variance (σ^2_{att}), are obtained by SA map Ω_{att} using features F_X, $X \in C, S_0, S_1$.

inconsistency between the stylized results generated from the style images with the same style pattern. The proposed style inconsistency loss $\mathcal{L}_{\text{SI}}$ can be formulated as follows:

$$\mathcal{L}_{\text{SI}}(C, S_0, S_1)|_{S_0, S_1 \in S} = \frac{\mathcal{L}_{\text{style}}(\theta(S_0, C), \theta(S_1, C))}{\mathcal{L}_{\text{style}}(S_0, S_1)} \tag{5}$$

where $\mathcal{L}_{\text{SI}}$ considers not only the style inconsistency between $\theta(S_0, C)$ and $\theta(S_1, C)$ but also the divergence between S_0 and S_1. We use $\mathcal{L}_{\text{style}}(S_0, S_1)$ as the denominator to count the influence of divergence from style images.

Including the proposed style inconsistency loss $\mathcal{L}_{\text{SI}}$, the final loss of our CAST framework is established as follows:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{content}} + \beta \mathcal{L}_{\text{style}} + \gamma \mathcal{L}_{\text{SI}} \tag{6}$$

where α, β, and γ are the corresponding parameters to highlight the importance of each individual loss. Thus, our CAST framework has an overall consideration of the content, style, and style consistency at the same time.

D. Plug-and-Play SA Module

In this section, we first review the statistical calculation of AST models and then propose a SA module for AST to calculate the SA weighted statistical features.

Given the pretrained VGG lth layer features $F \in \mathbb{R}^{C \times H \times W}$, the statistical features, mean (μ), and standard deviation (σ)

are defined as follows:

$$\mu(F) = \sum_{i,j} w_{i,j} F_{i,j} \tag{7}$$

$$\sigma(F) = \sqrt{\sum_{i,j} w_{i,j} (F_{i,j} - \mu(F))^2} \tag{8}$$

where $w_{i,j}$ denotes the weight in spatial domain, which is equal to $(1/(i \times j))$. It means that each element in the feature maps contributes equally. The statistical calculation of feature maps has been widely applied in AST [12], [14], [17]. For example, the AdaIN employees the mean and standard deviation of style features to normalize the content features in a holistic way. However, they treat all elements in feature maps equally and fail to consider the attention weights in spatial domain. This leads to unstable statistics, ignores the local style distribution, and thus may result in unsatisfying ST performance. Therefore, many works have been developed to address this issue. SAN [17] develops a style attention module and performs the feature normalization with the attention map generated by style and content. Furthermore, AdaAttN [19] proposes an adaptive attention module to normalize the content using a style-content attention map at the point-wise level. SAFIN [20] learns the factorized spatial normalization parameters for stylization from content and style images. Different from these above-mentioned methods, we propose a plug-and-play SA module to extract important local and global style patterns. As shown in Fig. 5, the proposed SA module is a shareable nonlocal structure. We adopt the nonlocal structure to generate an attention map which guides the calculation of the SA weighted statistics. To be specific, given the VGG lth layer feature $F_X \in \mathbb{R}^{C \times H \times W}$, a siamese 1×1 convolution layer with shareable weights is first leveraged to embed the feature into a new deep subspace $Y_X \in \mathbb{R}^{C \times HW}$. l_2 normalization "l_2 norm" in Fig. 5) is used to obtain the normalized feature $\widetilde{Y}_X$ which maps the feature to the multidimensional unit space

$$\widetilde{Y}_X = \frac{Y_X}{\|Y_X\|_2} \tag{9}$$

where $\|\cdot\|_2$ denotes the l_2-norm. Similar to the nonlocal structure in [33], we obtain the transposed feature $\widetilde{Y}_X^T$ for matrix multiplication. Following with the channel-wise average and softmax function, the SA weights $\Omega^{\text{att}} \in \mathbb{R}^{HW \times 1}$ can be acquired as:

$$\Omega_{\text{att}} = \text{Softmax}\big(M\big(\widetilde{Y}_X \otimes \widetilde{Y}_X^T\big)\big) \tag{10}$$

where $M(\cdot)$ denotes the average function along the spatial dimension, $\otimes$ denotes the matrix multiplication. Finally, the SA weighted mean (μ_{att}) and standard deviation (σ_{att}) are calculated as

$$\mu_{\text{att}}(Z_X) = Z_X \otimes \Omega_{\text{att}} \tag{11}$$

$$\sigma_{\text{att}}(Z_X) = \sqrt{(Z_X - \mu_{\text{att}}(Z_X))^2 \otimes \Omega_{\text{att}}} \tag{12}$$

where $Z_X \in \mathbb{R}^{C \times HW}$ indicates the embedded convolutional results of feature maps F_X in a new deep subspace. Using the proposed CAST framework, the obtained SA weighted statistics can well represent the key style features. We adopt a shareable nonlocal structure to reduce trainable parameters and calculate the auto-correlated attention weights. Different from the average weight, the SA weighted statistics can catch global attention in the long range of image features that capture the important style patterns. In addition, the proposed module can be plugged into the existing AST models and replace the non-attention weighted statistical calculation. With this plug-and-play SA module, the representative statistical features become robust and perform consistent ST.

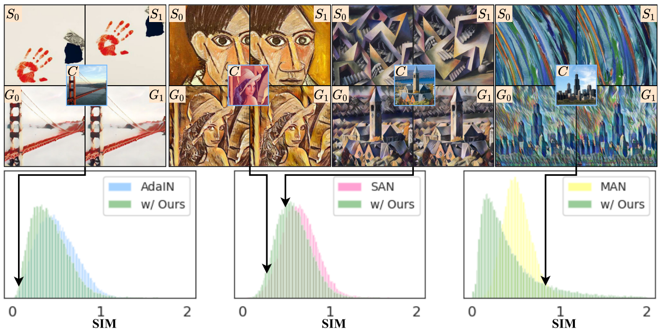

Fig. 6. Distributions of SIM for the proposed methods. In each pair, C denotes the content image, S_0 and S_1 denote the style images, and G_0 and G_1 denote the corresponding generated images. A larger value of SIM means higher inconsistency of the evaluated AST methods.

In Fig. 6, we visualize the SIM distributions of the AST models using the proposed CAST framework. Compared with Fig. 2, we can find that the consistency of the AST models with our CAST framework is greatly improved while maintaining the stylization performance. The SIM value tends to be distributed in the left area. It is also proved that SIM can well measure the style inconsistency of the AST model.

V. Experiments

In this section, we will present the comparative experiments with other state-of-the-art methods and discuss the qualitative and quantitative results.

A. Experiment Settings

In our experiments, the MS-COCO datasets [34] are used as the content images and WikiArt [35] as the style images to train the AST models. In the training phase, all images are transformed into size of 512 along the smaller dimension to well preserve the aspect ratio and then randomly cropped to size of 256 × 256. All test images with any input size are available for our consistent ST model. The Adam optimizer [36] are employed with learning rate of 10^{-4} at a decay rate of 5×10^{-5} in each iteration. We use the pre-trained VGG-19 as our encoder to extract deep features. We train all models on the Pytorch-1.6.0 [37] and NVIDIA Quadro P6000.

B. Performance Comparison

To validate the performance, we compare our consistent approaches with some state-of-the-art AST methods including AdaIN [12], LST [16], SAN [17], MAN [18], DIN [14], SAFIN [20], AdaAttN [19], StyleFormer [27], ST with transformers (StyTr2) [28], and exact feature distribution matching (EFDM) [26].

1) Qualitative Evaluation: Compared with other state-of-the-art methods, Fig. 7 shows the visual results of ST using different content and style images. As can be seen, AdaIN brings up the unwanted style patterns such as color noisy. LST can well preserve the original content structure. However, the generated images lose some important style patterns of the original style images. SAN generates abundant style patterns but cannot align them well with content images, resulting in style mixture. MAN alleviates the style mixture but still pays little attention to the essential feature patterns in style images so as to bring more unimportant style patterns into the generated results. As to SAFIN, the style patterns are generated well but do not match the content image. This causes a detailed content structure corruption. As shown in the last third row of Fig. 7, the stylization results generated by StyleFormer exhibit semantic content coherency, but they also contain some color styles that are not present in the input style image. StyTr2 achieves impressive stylization performance by incorporating style patterns into content images with long-range dependency. However, the corresponding row in Fig. 7 shows that it generates results with unpleasant color noise. EFDM employs feature distribution matching for ST but unfortunately results in a degradation of content structure and an unsatisfactory stylization effect. In contrast, our method can generate essential style patterns from style images and achieve promising ST performance.

To evaluate the style consistency of ST models, Fig. 8 presents the visual results of style consistency. "CAST" denotes the proposed CAST framework. For example, "CAST-SAN" means the CAST method using SAN as the feature transformation module. We can clearly observe that the proposed methods using the CAST framework can well catch the important style patterns (color, shape, and texture) from style images and obtain a robust stylization result. In detail, SAN stylizes the content images with different color and shape patterns in the human face and eyes, resulting shape inconsistencies as depicted in the blue rectangle box of the subfigure. CAST-SAN can align important style patterns well in the generated images without shape artifacts. MAN generates the same shape patterns in the stylized images. However, the color patterns vary a lot between the generated images, especially in the red rectangle area. CAST-MAN can extract key style patterns from the style images and obtain robust stylization results with better color consistency (the same color of human and background). AdaIN generates inconsistent stylization results with content blurred. As shown in the red rectangle area, it smooths the bridge line resulting in content inconsistency. With the proposed method, CAST-AdaIN can obtain the consistent ST (the same color and shape in bridge line). For style images with different color weights, LST transfers the color without attention based weights in the style image to obtain inconsistent results (the different cloud color as shown in red rectangle). CAST-LST focuses on extracting important style patterns (hair color and shape in the style images) and obtains more consistent results in the generated images (the same cloud color and shape) as shown in the blue rectangle box. The stylization results of StyTr2 exhibit color inconsistency between the style image pairs, which adversely affects the overall quality of the generated images. In contrast, CAST-StyTr2 can effectively capture crucial style features and maintain consistency during the ST process. The generated results by EFDM present inconsistent color and texture patterns in the hair of the girl due to the neglect of essential style patterns, leading to inconsistent ST. Compared to EFDM, CAST-EFDM generates images with essential color and texture patterns that are consistently transferred into the content images. It is worth noting that the visual consistency comparison results align with the SIM shown in Fig. 8.

2) Quantitative Evaluation: To evaluate the performance of ST methods, we use the deception rate (DR) [38], balanced style loss (BSL) [39], structural similarity index measure (SSIM) [40], and SIM to verify the ability of style transferring (DR and BSL), content preservation (SSIM), and style consistency transferring (SIM), separately.

For testing the style transferability, we calculate the DR as in [38]. It is calculated as the success rate of generated images to deceive the fine-tuned classification model such that both the style and generated images are predicted as the same category. The higher DR means the better style transferability. We employ the ArtImage in [35]. It consists of 9000 images with five art categories (drawings, engraving, iconography, painting, and sculpture) and the Microsoft COCO 2014 (MS-COCO) dataset [34] as the style and content images. As for dataset split, we follow the training and testing split as the ratio of 7:1. We fine-tune the pre-trained VGG19 as our classification model using the training split. We randomly select 1000 style-content pairs as the validation set for AST. Besides, we also calculate the BSL [39] to evaluate the ability of ST. Lower BSL means better style transferability of the model. We calculate the SSIM [40] between generated and content images to measure the ability of content preservation. To evaluate the style consistency quantitatively, we calculate the SIM of ST methods defined in (3). The style images are divided into four parts equally and fed into ST models as the style images, separately. The higher SIM means the more serious inconsistency, and vice versa.

Table I presents the quantitative results of the ST models. "Imp." and "RImp." denote absolute and relative improvement, respectively. As shown in the results, with the proposed CAST framework, the performance of ST and consistency can be improved effectively while well preserving the content structure. This is consistent with the qualitative results in Figs. 7 and 8. In terms of DR, BSL, SSIM, and SIM, **CAST-AdaIN** obtains the relative improvements of 20.80%, 74.32%, 0.93%, and 14.29%. Compared with SAN, **CAST-SAN** achieves the relative improvements of 4.56%, 20.17%, 3.67%, and 30.22%. **CAST-MAN** obtains the relative improvements of 19.29%, 23.68%, 3.00%, and 11.11%. **CAST-LST** achieves the relative improvements of 37.46%, 50.74%, 1.29%, and 5.88%. Using the proposed methods, **CAST-StyTr2** and **CAST-EFDM** can achieve significant improvements of 11.65%, 3.53%, 4.32%, 13.04% and 4.05%, 3.74%, 2.16%, 28.33% over StyTr2 and EFDM. These results strongly suggest that the ST methods employing our proposed consistent framework are capable of effectively capturing the essential style patterns, including

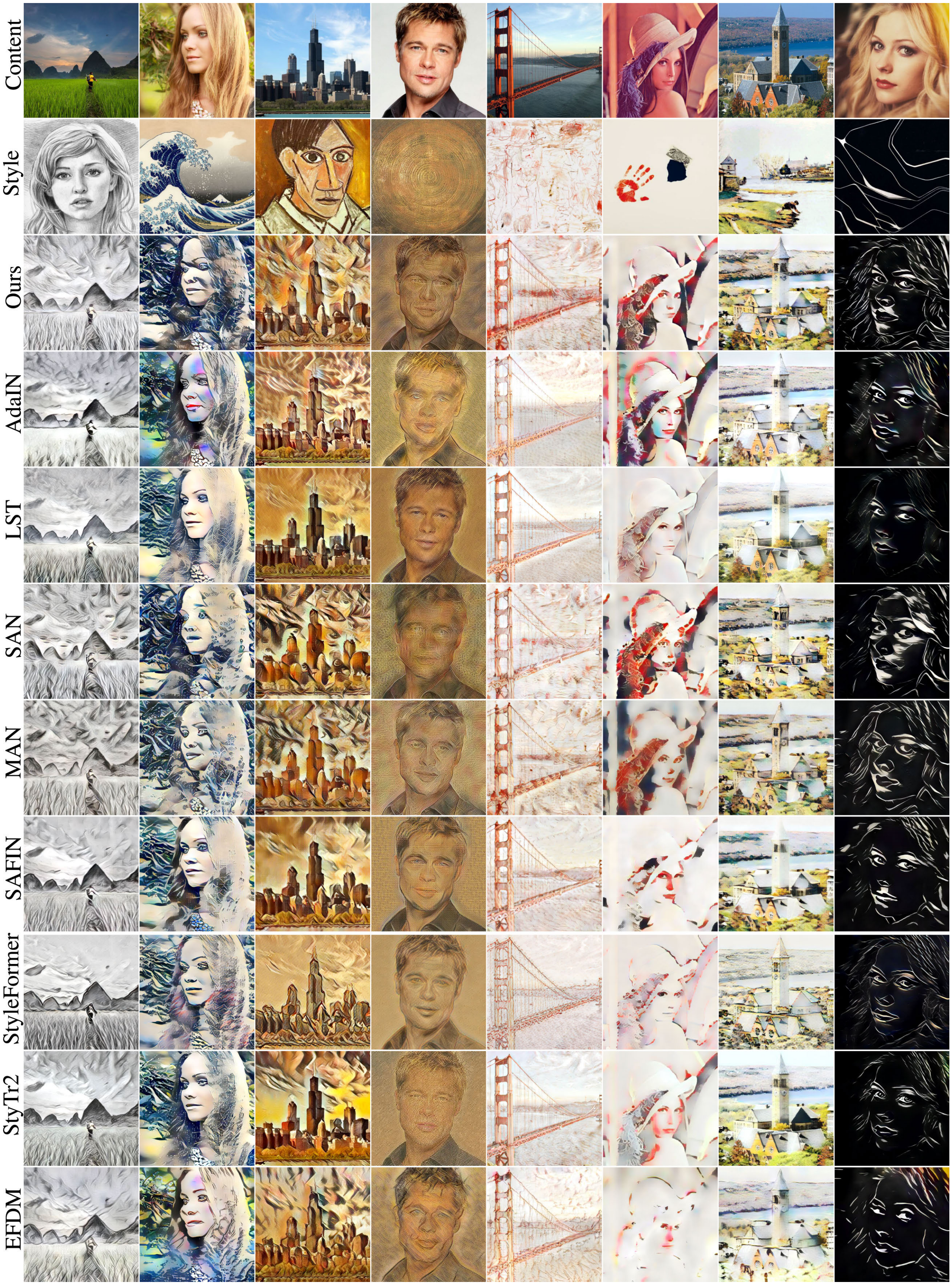

Fig. 7. Stylization results of competing methods. From left to right, each column shows the content images, style images, and images generated by AdaIN [12], LST [16], SAN [17], MAN [18], SAFIN [20], StyleFormer [27], StyTr2 [28], and EFDM [26]. As can be seen, our proposed method can generate the key style patterns and achieve satisfactory performance. The figure is viewed better in zoomed-in view version.

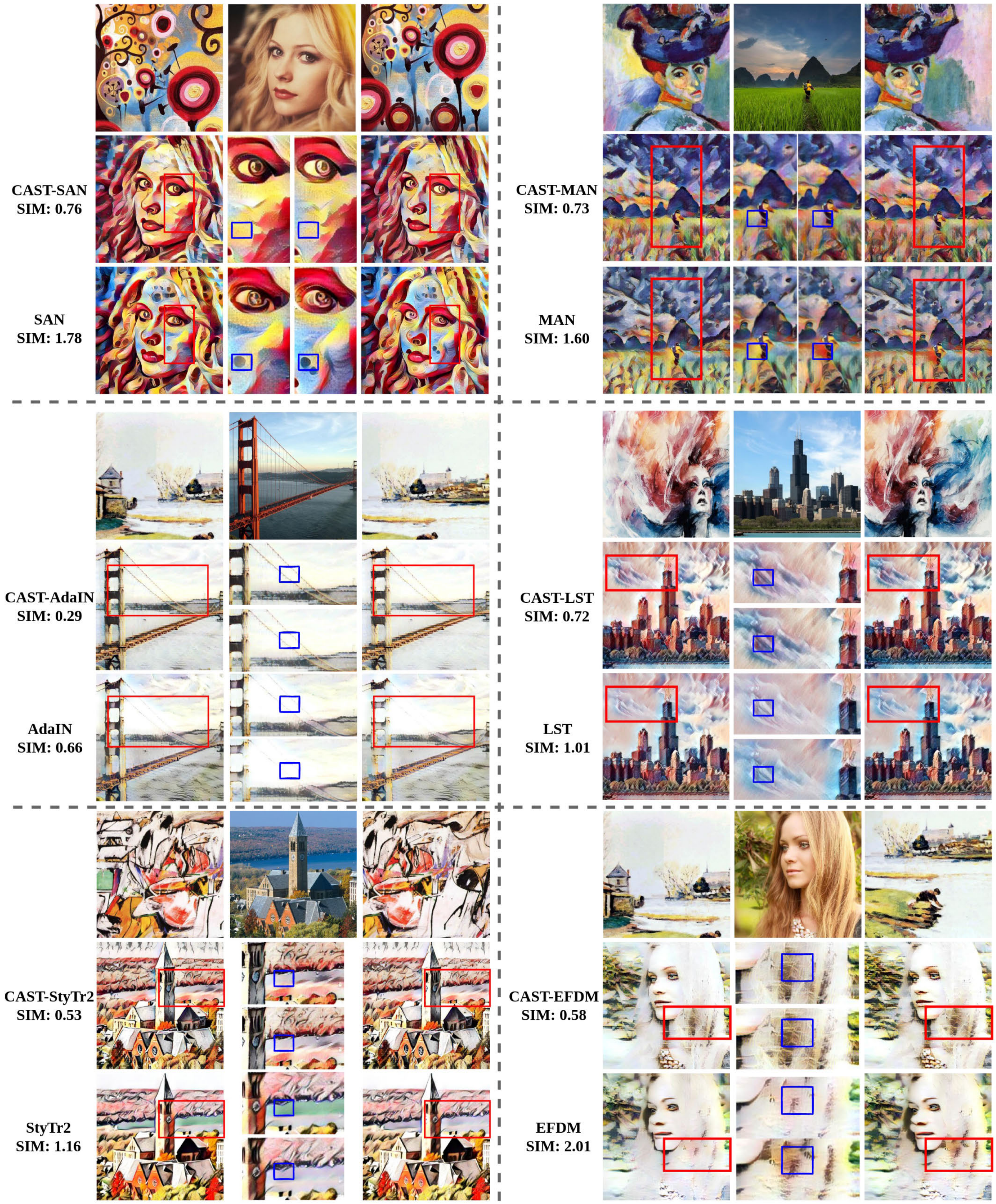

Fig. 8. Visual comparison of several consistent ST methods, namely SAN [17], MAN [18], AdaIN [12], LST [16], StyTr2 [28], and EFDM [26]. “CAST-” denotes the proposed CAST framework using the corresponding AST method within the feature transformation module. Each subfigure consists of three rows: the first row shows the content and style images, the second row displays the results generated by the proposed consistent method, and the third row displays the generated images by the competing methods. Our consistent methods exhibit superior performance in terms of color and shape consistency, with generated images containing richer style patterns and fewer artifacts compared to the competing methods.

TABLE I

PERFORMANCE COMPARISON OF ST MODELS

Method	**DR (%) ↑**	**BSL ↓**	**SSIM (%) ↑**	**SIM ↓**
AdaIN [12]	37.83%	4.44	32.41%	0.49
CAST-AdaIN	**45.70%**	**1.14**	**32.71%**	**0.42**
Imp.	7.87%	-3.30	0.30%	-0.07
RImp. (%)	20.80%	-74.32%	0.93%	-14.29%
SANet [17]	46.70%	2.33	30.82%	1.39
CAST-SAN	**48.83%**	**1.86**	**31.95%**	**0.97**
Imp.	2.13%	-0.47	1.13%	-0.42
RImp. (%)	4.56%	-20.17%	3.67%	-30.22%
MANet [18]	43.55%	3.59	29.97%	0.99
CAST-MAN	**51.95%**	**2.74**	**30.87%**	**0.88**
Imp.	8.40%	-0.85	0.90%	-0.11
RImp. (%)	19.29%	-23.68%	3.00%	-11.11%
LST [16]	36.52%	2.72	37.95%	0.51
CAST-LST	**50.20%**	**1.34**	**38.44%**	**0.48**
Imp.	13.68%	-1.38	0.49%	-0.03
RImp. (%)	37.46%	-50.74%	1.29%	-5.88%
StyTr2 [28]	31.84%	1.70	45.86%	0.46
CAST-StyTr2	**35.55%**	**1.64**	**47.84%**	**0.40**
Imp.	3.71%	0.06	1.98%	-0.06
RImp. (%)	11.65%	-3.53%	4.32%	-13.04%
EFDM [26]	38.48%	1.87	29.20%	0.60
CAST-EFDM	**40.04%**	**1.80**	**29.83%**	**0.42**
Imp.	1.56%	-0.07	0.63%	-0.17
RImp. (%)	4.05%	-3.74%	2.16%	-28.33%
StyleFormer [27]	39.45%	4.80	31.79%	0.69
DIN [14]	41.02%	1.77	30.47%	0.51
SAFIN [20]	42.58%	1.90	40.39%	0.90
AdaAttN [19]	35.35%	5.71	47.09%	1.08

Note: The **boldface** denotes the proposed methods using our consistent arbitrary style transfer (CAST) framework. "CAST-" denotes the proposed CAST framework using the corresponding arbitrary style transfer method within the feature transformation module.

TABLE II

ABLATION STUDY OF DIFFERENT CONSISTENCY TRAINING METHODS

Method	**BSL ↓**	**SSIM (%) ↑**	**SIM ↓**
Baseline	4.44	32.41%	0.49
Baseline-IoUPC	2.60	34.94%	0.47
Imp.	-1.84	2.53%	-0.02
RImp. (%)	-41.44%	7.81%	-4.08%
Baseline-SILR	3.48	32.88%	0.41
Imp.	-0.96	0.47%	-0.09
RImp. (%)	-21.62%	1.45%	-18.36%
Baseline-SCT	**2.54**	**32.50%**	**0.40**
Imp.	-1.90	0.09%	-0.09
RImp. (%)	-42.79%	0.28%	-19.37%
Baseline-UDA [32]	2.70	31.33%	0.42
Imp.	-1.74	-1.08%	-0.07
RImp. (%)	-39.19%	-3.33%	-14.29%
Baseline-RSL [21]	6.29	36.00%	0.43
Imp.	1.85	3.59%	-0.06
RImp. (%)	41.67%	11.08%	-12.24%

Note: Baseline refers to the CAST-AdaIN without IoUPC, SILR, and SA. SCT denotes the style consistency training including IoUPC and SILR. UDA means the unsupervised data augmentation [32]. RSL refers to the relaxed style loss [21].

color and texture, and consistently transfer them into content images, resulting in remarkable consistent ST performance.

VI. Ablation Study

In this section, we conduct experiments to discuss the functions of each component and verify the importance of the proposed methods.

A. Discussion of SCT

Table II presents the experimental results of different consistent models. AdaIN is a well-established method for AST and is frequently used as a baseline for developing new ST methods. Therefore, we adopt AdaIN as the feature transformation module within our proposed CAST framework (called CAST-AdaIN) for our baseline approach. Baseline refers to the proposed CAST-AdaIN without the proposed IoU-preserving crop (IoUPC) module and style inconsistency loss regularization (SILR) term. We extend the Baseline network by integrating different modules to evaluate their effectiveness, denoted as "Baseline-". For example, "Baseline-IoUPC" refers to the Baseline network with IoUPC module. **SCT** denotes the proposed style consistency training method including the proposed IoUPC module and SILR term. In **SCT**, the IoUPC module and SILR term work together to enable consistent style training in the CAST framework. Specifically, the IoUPC module is employed to obtain style image pairs with minor disturbance from the input style and the SILR term facilitates the network in learning consistent style features. In the experiments, some consistency training methods such as unsupervised data augmentation (UDA) [32] and relaxed style loss (RSL) [21] are also compared. In the comparison of "UDA" method, we adopt the same manipulation in [32] for a fair comparison. "RSL" relaxes the objective function to make the style loss term more robust. In a robust ST model, we would like consistently transfer important style patterns into content images. Therefore, we use the evaluation metrics of BSL [39], SSIM [40], and SIM to evaluate the ability of ST, content preservation, and style consistency. The results indicate that Baseline-IoUPC effectively generates stylization results and Baseline-SILR optimizes the network to consistently transfer style features into content images. By integrating IoUPC and SILR, the proposed **Baseline-SCT** achieves significant relative improvements of 42.79%, 0.28%, and 19.37% in terms of BSL, SSIM, and SIM, respectively, compared to the Baseline network. These demonstrate that the proposed **SCT** approach can facilitate the learning of essential style features by the network and consistently transfer style patterns into content images to achieve significant improvements of consistent stylization. Compared to the original method, the proposed consistent ST method not only captures richer style patterns from the style image but also retains the content structure of the content image. UDA simply minimizes the difference between the generated images to obtain the style patterns directly. However, it fails to fully consider the minor variation between style images resulting in content distortion. On the contrary, RSL can retain more content structure to resist interference from content to achieve consistent ST but fails to effectively extract the important style patterns. **SCT** can well capture the important style patterns in the style image and preserve the content structure at the same time to achieve consistent ST.

TABLE III

ABLATION STUDY OF DIFFERENT ATTENTION MECHANISMS

Method	BSL ↓	SSIM (%) ↑	SIM ↓
Baseline	4.44	32.41%	0.49
Baseline-SA	**1.14**	**32.71%**	**0.42**
Imp.	-3.30	0.30%	-0.07
RImp. (%)	-74.32%	0.93%	-14.29%
Baseline-MLP	1.35	32.34%	0.48
Imp.	-3.09	-0.07%	-0.01
RImp. (%)	-69.59%	0.22%	-2.04%
Baseline-SAWC	1.70	28.39%	0.44
Imp.	-2.74	-4.02%	-0.05
RImp. (%)	-61.71%	-12.40%	-10.20%
Baseline-SAWR	1.57	28.16%	0.50
Imp.	-2.87	-4.25%	0.01
RImp. (%)	-64.64%	-13.11%	2.04%
Baseline-SN [41]	2.81	32.15%	0.48
Imp.	-1.63	-0.26%	-0.01
RImp. (%)	-36.71%	-0.80%	-2.04%
Baseline-AN [30]	5.68	31.87%	0.46
Imp.	1.24	-0.54%	-0.03
RImp. (%)	27.93%	-1.67%	-6.12%

Note: Baseline refers to the CAST-AdaIN without IoUPC, SILR, and SA. SA denotes the proposed self-attention module. MLP refers to multiple-layer-perceptron network. SAWC and SAWR means the proposed self-attention module without convolution and with ReLU layer after convolution in attention learning. SN and AN denote the swithable and attentive normalization.

B. Analysis of Attention Module

In Table III, we evaluate different attention-based methods to provide an optimal module for CAST. **SA** denotes our designed SA module as shown in Fig. 5. MLP means the multiple layer perception with several convolutional neural network layers to learn the attention maps. In contrast, our proposed SA module adopts SA mechanisms to capture the inner-correlations between feature maps. To explore the optimal network structure as shown in Fig. 5, the proposed SA structure without convolution and with ReLU layers, denoted by SAWC and SAWR, are compared in the ablation studies. To validate the efficacy of the proposed SA module, some weighted normalization methods such as switchable normalization (SN) and attentive normalization (AN) are also compared. **Baseline-SA** effectively captures essential style features, leading to significant improvements over the "Baseline" method in terms of BSL, SSIM and SIM by 74.32%, 0.93% and 14.29%. Using a simple network, the Baseline-MLP learns initial features and achieves modest improvements compared to the proposed method. By exploring self-attentive features, our Baseline-SA outperforms Baseline-MLP by 15.56% in BSL, 0.53% in SSIM, and 12.5% in SIM. These demonstrate that the proposed SA module can effectively capture essential style features, thereby enhancing the CAST performance. As shown in the row "Baseline-SAWC," **SA** without the convolution layer obtains the improvement of ST and consistency but lacks the representative ability to preserve the content structure. As shown in the row "Baseline-SAWR," **SA** with ReLU layer ignores some important activation from the generated SA weights. Therefore, the Baseline-SAWR performs an inconsistent AST, resulting in a degradation in SIM. Baseline-SN enriches style patterns by adapting the normalization method into the ST method according to the inputs but fails to preserve the content structure. Baseline-AN learns the attention weights to conduct the mixture normalization. It is effective for object classification but not suitable to improve the consistency of ST. Overall, the proposed **SA** devises an optimal SA module to capture the important features for consistent ST and achieves the best performance in terms of BSL, SSIM, and SIM, respectively.

VII. CONCLUSION

In this article, we first discovered the style inconsistency problem in the AST model and developed a measure, namely SIM, to quantitatively evaluate the inconsistency. To address this issue, we then proposed a CAST framework that consists of an IoUPC module, a SILR, and a SA module to capture important features for consistent stylization. We conducted comprehensive qualitative and quantitative experiments to verify the effectiveness of the proposed approach. The results demonstrate that our methods significantly improve the consistency of stylization and enhance the stylization performance by capturing salient style features. In future work, we plan to extend our consistent model to video ST and other image generation tasks.

ACKNOWLEDGMENT

This work was done prior to Amazon involvement of the authors.

REFERENCES

[1] J. J. Virtusio, J. J. M. Ople, D. S. Tan, M. Tanveer, N. Kumar, and K.-L. Hua, "Neural style palette: A multimodal and interactive style transfer from a single style image," *IEEE Trans. Multimedia*, vol. 23, pp. 2245–2258, 2021.

[2] Z. Ma et al., "Dual-affinity style embedding network for semantic-aligned image style transfer," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 2, 2022, doi: 10.1109/TNNLS.2022.3143356.

[3] S. Zhao, L. Zhang, Y. Shen, and Y. Zhou, "RefineDNet: A weakly supervised refinement framework for single image dehazing," *IEEE Trans. Image Process.*, vol. 30, pp. 3391–3404, 2021.

[4] K. Xu, L. Wen, G. Li, H. Qi, L. Bo, and Q. Huang, "Learning self-supervised space-time CNN for fast video style transfer," *IEEE Trans. Image Process.*, vol. 30, pp. 2501–2512, 2021.

[5] X. Xiao, Y. Chen, Y.-J. Gong, and Y. Zhou, "Prior knowledge regularized multiview self-representation and its applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1325–1338, Mar. 2021.

[6] Y. Zhang, Y. Wang, X. Chen, X. Jiang, and Y. Zhou, "Spectral–spatial feature extraction with dual graph autoencoder for hyperspectral image clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 12, pp. 8500–8511, Dec. 2022.

[7] R. Ma, B. Zhang, Y. Zhou, Z. Li, and F. Lei, "PID controller-guided attention neural network learning for fast and effective real photographs denoising," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 3010–3023, Jul. 2022.

[8] 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.

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

[10] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 694–711.

[11] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "StyleBank: An explicit representation for neural image style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2770–2779.

[12] 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. 1510–1519.

[13] Z. Zhou, Y. Wu, X. Yang, and Y. Zhou, "Neural style transfer with adaptive auto-correlation alignment loss," *IEEE Signal Process. Lett.*, vol. 29, pp. 1027–1031, 2022.

[14] Y. Jing et al., "Dynamic instance normalization for arbitrary style transfer," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 4369–4376.

[15] 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. 386–396.

[16] X. Li, S. Liu, J. Kautz, and M.-H. Yang, "Learning linear transformations for fast image and video style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3804–3812.

[17] 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. 5873–5881.

[18] 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.

[19] 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. 6629–6638.

[20] A. Singh, S. Hingane, X. Gong, and Z. Wang, "SAFIN: Arbitrary style transfer with self-attentive factorized instance normalization," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2021, pp. 1–6.

[21] 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.

[22] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua, "Stereoscopic neural style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6654–6663.

[23] Y. Yao, J. Ren, X. Xie, W. Liu, Y.-J. Liu, and J. Wang, "Attention-aware multi-stroke style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1467–1475.

[24] J.-R. Chang and Y.-S. Chen, "Exploiting spatial relation for reducing distortion in style transfer," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 1208–1216.

[25] Y. Lu, Z. Zhang, G. Lu, Y. Zhou, J. Li, and D. Zhang, "Addi-Reg: A better generalization-optimization tradeoff regularization method for convolutional neural networks," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10827–10842, Oct. 2022.

[26] Y. Zhang, M. Li, R. Li, K. Jia, and L. Zhang, "Exact feature distribution matching for arbitrary style transfer and domain generalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8025–8035.

[27] 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. 14598–14607.

[28] Y. Deng et al., "StyTr2: Image style transfer with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11316–11326.

[29] Z. Zhou, Y. Chen, and Y. Zhou, "Deep dynamic memory augmented attentional dictionary learning for image denoising," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Feb. 27, 2023, doi: 10.1109/TCSVT.2023.3249796.

[30] X. Li, W. Sun, and T. Wu, "Attentive normalization," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 70–87.

[31] Z. Zhou and Y. Zhou, "Cross-channel similarity based histograms of oriented gradients for color images," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 1621–1625.

[32] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1–13.

[33] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

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

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

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[37] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Dec. 2019, pp. 8026–8037.

[38] 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.

[39] J. Cheng, A. Jaiswal, Y. Wu, P. Natarajan, and P. Natarajan, "Style-aware normalized loss for improving arbitrary style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 134–143.

[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[41] P. Luo, J. Ren, Z. Peng, R. Zhang, and J. Li, "Differentiable learning-to-normalize via switchable normalization," *Int. Conf. Learn. Represent.*, 2019, pp. 1–19.

Zheng Zhou received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2017, and the Ph.D. degree from the Department of Computer and Information Science, University of Macau, Macau, China, in 2023.

His research interests include computer vision, image processing, feature extraction, and deep learning.

Yue Wu (Member, IEEE) received the B.E. degree in telecommunication engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2001, the M.S. degree in applied mathematics from the University of Toledo, Toledo, OH, USA, in 2008, and the Ph.D. degree in electrical engineering from Tufts University, Medford, MA, USA, in 2012.

He was a Computer Scientist with the Information Sciences Institute, University of Southern California Viterbi School of Engineering, Los Angeles, CA, USA. He is currently a Principal Applied Scientist with the Amazon Alexa Natural Understanding, Manhattan Beach, CA, USA. His research focuses on information security, image processing, and pattern recognition.

Yicong Zhou (Senior Member, IEEE) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, in 1992, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA, in 2008 and 2010, respectively.

He is a Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include image processing, computer vision, machine learning, and multimedia security.

Dr. Zhou is a fellow of the Society of Photo-Optical Instrumentation Engineers (SPIE) and was recognized as one of the "Highly Cited Researchers" in 2020 and 2021. He serves as an Associate Editor for the IEEE Transactions on Cybernetics, IEEE Transactions on Neural Networks and Learning Systems, and IEEE Transactions on Geoscience and Remote Sensing.