Pluggable Style Representation Learning for Multi-Style Transfer

Hongda Liu[1], Longguang Wang[2], Weijun Guan[1], Ye Zhang[1], and Yulan Guo[1]

[1] The Shenzhen Campus of Sun Yat-Sen University, Sun Yat-Sen University
[2] Aviation University of Air Force
{liuhd36@mail2.sysu,guoyulan@sysu}.edu.cn

Abstract. Due to the high diversity of image styles, the scalability to various styles plays a critical role in real-world applications. To accommodate a large amount of styles, previous multi-style transfer approaches rely on enlarging the model size while arbitrary-style transfer methods utilize heavy backbones. However, the additional computational cost introduced by more model parameters hinders these methods to be deployed on resource-limited devices. To address this challenge, in this paper, we develop a style transfer framework by decoupling the style modeling and transferring. Specifically, for style modeling, we propose a style representation learning scheme to encode the style information into a compact representation. Then, for style transferring, we develop a style-aware multi-style transfer network (SaMST) to adapt to diverse styles using pluggable style representations. In this way, our framework is able to accommodate diverse image styles in the learned style representations without introducing additional overhead during inference, thereby maintaining efficiency. Experiments show that our style representation can extract accurate style information. Moreover, qualitative and quantitative results demonstrate that our method achieves state-of-the-art performance in terms of both accuracy and efficiency. The codes are available in https://github.com/The-Learning-And-Vision-Atelier-LAVA/SaMST.

Keywords: Style Transfer · Pluggable Style Representation · Efficient Style Generation

1 Introduction

Style transfer (ST) aims at capturing image style to generate artistic images, which has attracted increasing interests since the seminal works [12, 13]. Style extracting ability and generation quality are two critical research directions in the area of style transfer. In early years, a single model only accomplishes single style transfer (SST) [23, 37, 38]. To improve the flexibility of ST, multi-style transfer (MST) methods aim to incorporate multiple styles into one single model [3, 10, 55]. Recently, arbitrary-style transfer (AST) methods are proposed for wider style domain [2, 9, 17, 19, 22, 27, 33, 54, 56, 59].

Under sufficient computational resources, AST methods are able to handle a wide style domain. However, in real applications, deploying ST models on edge devices is highly demanded. Existing AST methods commonly suffer large model sizes and low inference efficiency, which hinders them to be deployed on resource-limited devices.

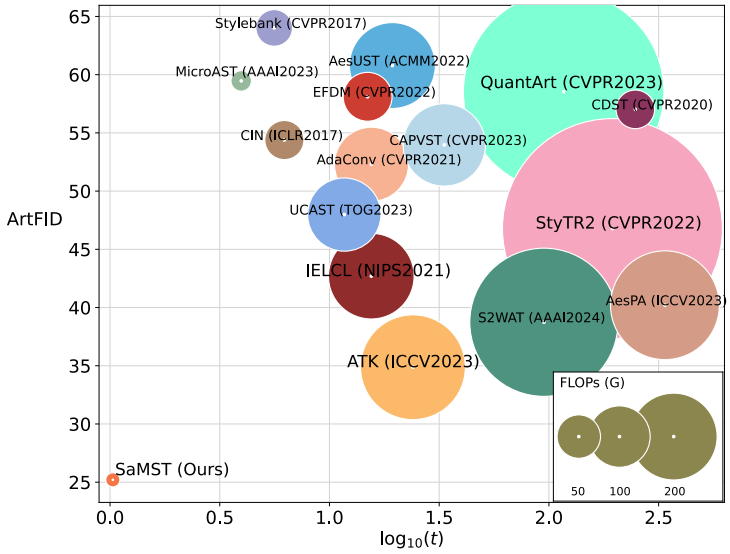

Fig. 1: Trade-off between inference time t (ms) and ArtFID [48] achieved by different methods. The size of a circle represents FLOPs.

To remedy this, several efforts are made to develop lightweight network structures [22, 32, 46]. However, the limited model capacity of lightweight models hinders them to accommodate diverse styles, resulting inferior quality of generated images. To strike a balance between inference efficiency and generation quality, several methods [3, 10, 52, 55] are developed to store image styles in separate network structures. Although these methods achieve faster inference speed, their model size increases dramatically as style numbers increase, resulting in a huge storage burden. Moreover, these methods suffer from difficulties in extending to new styles [3, 10, 55].

To address the issues above, we introduce a pluggable style representation learning scheme. This scheme inherits the high inference efficiency and generation quality of MST while maintaining superior style capacity of AST. Specifically, we encode the style-specific information into a compact representation and store it in a style codebook (SCB). Moreover, we propose a style-aware multi-style transfer (SaMST) network with flexible adaption to different styles based on the learned representations. Particularly, our SaMST incorporates style information to perform feature adaption by predicting convolutional kernels, variance/mean values and channel-wise modulation coefficients from the compact style representation. Our SaMST can reduce the model size and computational complexity while improving the quality of results, striking a better balance between accuracy and efficiency, as illustrated in Fig. 1. In addition, we propose an incremental style extension scheme. This scheme enables our model to quickly adapt to a new style representation without forgetting the previous styles. Experiments show our SaMST not only produces visually more pleasing results (Fig. 2 and Fig. 4), but also achieves over $4\times$ reduction in model size and over $3\times$ speedup for each style (Table 1).

In summary, our contributions are three-fold:

Fig. 2: An 2K stylized sample (2028×1440), rendered in about 0.01 seconds on a single NVIDIA RTX 3090 GPU. The upper left and down left images are the content and style images, respectively.

- We introduce a pluggable style representation learning scheme for MST. Moreover, we propose a style-aware multi-style transfer (SaMST) network with flexible adaption to different styles based on the learned representations. And our SaMST achieves great advantages in terms of efficiency.
- To solve the size explosion and style catastrophic forgetting of previous MST methods, we propose a novel style representation extension scheme which has stronger application advantages.
- Extensive experiments show that our method produces state-of-the-art results in terms of both visual quality and quantitative performance.

2 Related Work

2.1 Neural Style Transfer

In earlier stage, Gatys *et al.* [12, 13] propose optimization-based methods to obtain stylized images. However, numerous iterations are required to obtain a satisfied result. To achieve faster generation speed, feed-forward methods [23, 37, 38] are proposed. They force the whole network to learn a certain style, so that it achieves fast style transfer with any content image. Furthermore, researchers adapt multiple image styles to corresponding network structures [3, 10, 55] to enhance the generalization of the ST. Some researchers find that pre-trained VGG [35] can accurately capture image content and style information. So it was applied to style transfer as an image feature encoder. The scheme is capable of any image style [2, 9, 17, 19, 22, 27, 33, 56, 59]. With the wide application of attention mechanism and transformer structure in computer vision, some researchers utilize them to enhance stylized image quality [8, 49, 54]. Inspired by the developments of contrastive learning, several efforts [4, 46, 57, 58] have been made to

leverage contrastive learning to obtain better stylized results. In addition, as a practical image generation technique, the quality of style transfer [45, 48, 53], the diversity of style textures [28, 39, 44], user control [1, 14, 26] and other practical issues also attract researchers' concern.

In real applications, model efficiency and size are crucial. As the seminal work of feed-forward methods, [23] greatly improves the inference speed compared to optimization based methods. Doumoulin *et al.* [10] stores style-specific parameters into a learnable instance norm layer to reduce model size. Chen *et al.* [3] proposes stylebank to store a style in a certain module. Specific style module reduces inference time, at the cost of storage. MFS methods suffer from the problem of extreme model size inflation [3] or style catastrophic forgetting [10,55] when finetuning new styles. Researchers further design lightweight models [22, 32, 46] based on AST. With the developments of knowledge distillation [16], some methods compress the streamlined ST models [5, 41] from large pre-trained models. However, AST methods still face the problems of model redundancy, slow inference and weak perception in wider style domain.

2.2 Efficient Network Architecture

Inspired by prior research on transformation-invariant scattering [34], Laurent Sifre develops depthwise convolutions, and uses them in AlexNet to obtain small gains in accuracy and large gains in convergence speed, as well as a significant reduction in model size [40]. Then this designed layer is used in many classic efficient vision backbones (*e.g.*, Inception V1 and V2 [20, 36], Xception [6] and MobileNets [18]). In recent years, depthwise convolutions are applied in some practical tasks to improve inference efficiency, such as image super-resolution [42, 50], neural machine translation [24] and style transfer [2]. Besides, Jin *et.al* [21] also propose a flattened network that consist of consecutive sequence of one-dimensional filters across all directions in 3D space, which obtains comparable performance as conventional convolutional networks and reduces model size. Wang *et.al* [43] propose to factorize the convolutional layer to reduce its computation.

3 Methodology

3.1 Overview

Our multi-style style transfer framework consists of style-aware multi-transfer (SaMST) network and style codebook (SCB), as shown in Fig. 3(a). First, the content image c is fed to Encoder to obtain content feature E. Then, a style representation (*e.g.*, $f_i \in \mathbb{R}^C$ and $C = 16$) is selected as style condition information from SCB, which is employed to adapt the generator parameters. Next, E is fed to the generator to obtain stylized image feature E_i. Finally, E_i is decoded by the decoder to obtain stylized image I_i. Note that, our framework is compatible to diverse encoder and decoder architectures. For real applications (*i.e.*, efficiency and model size), a lightweight 3-layer symmetric encoder and decoder is employed in our framework, which is similar to [3, 10, 51].

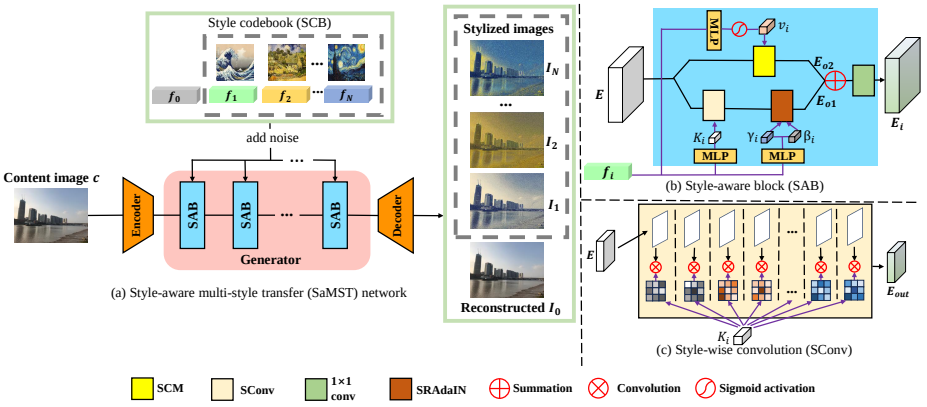

Fig. 3: An overview of our multi-style transfer framework.

3.2 Style-aware Multi-style Transfer (SaMST) Network

Style-wise Convolution (SConv) To better preserve local geometric structures of style images, AdaConv [2] employs a style-aware depthwise convolution. Inspired by AdaConv, our SConv (Fig. 3(c)) layer learns to predict the kernel of a depthwise convolution conditioned on a style representation to achieve flexible style-aware adaption. Our SConv not only makes the network lightweight and flexible, but also captures more local geometric structures to promote stylized results.

Specifically, we obtain a style representation by explicitly selecting from the provided image styles (*e.g.*, f_i). Then, this representation is used as conditional information and passed to a MLP to generate the convolution kernels K_i in the style-aware block (Fig. 3(b)). Note that, $K_i \in \mathbb{R}^{C_{in}\times 1\times k_w\times k_h}$ and C_{in} represents channel number of the input content image feature E. Details of the style-wise convolution are shown in Fig. 3(c). The predicted convolution kernels perform depthwise convolution operation on content image feature E. Afterwards, the style-wise convolution is conducted as follows:

$$E_{out} = \text{Sconv}(K_i, E; groups) = K_i \otimes E, \tag{1}$$

where $groups$ is the number of convolution groups, and $\otimes$ denotes the depthwise convolution operation.

Style-representation Adaptive Instance Norm (SRAdaIN) In addition to local geometric structures, the global properties are also critical to the final results. Inspired by [19], we propose SRAdaIN to further capture global properties from style images. Specifically, the style representation f_i is fed to a MLP to predict the mean $\gamma_i \in \mathbb{R}^{C_{in}}$ and variance $\beta_i \in \mathbb{R}^{C_{in}}$ of the style. Afterwards, SRAdaIN is performed as follows:

$$E_{o1} = \text{SRAdaIN}(E_{out}, \gamma_i, \beta_i) = \beta_i\Big(\frac{E_{out} - \mu(E_{out})}{\sigma(E_{out})}\Big) + \gamma_i, \tag{2}$$

where $\mu(\cdot), \sigma(\cdot)$ refer to compute mean and variance of input features, respectively, which is similar to [10, 19].

Style-wise Channel Modulation (SCM): Inspired by CResMD [15] that uses controlling variables to rescale different channels to handle multiple image degradations, our SCM learns to generate modulation coefficients based on the style representation to perform channel-wise feature adaption. Specifically, the selected style representation f_i is passed to another MLP and a sigmoid activation layer to generate channel-wise modulation coefficients v_i. Then, v_i is used to rescale different channel components in E, obtaining E_{o2}.

$$E_{o2} = \mathrm{SCM}(E, v_i) = E \odot v_i, \tag{3}$$

where $\odot$ refers to element-wise multiplication with broadcast over the spatial dimensions. Finally, E_{o1} is summed up with E_{o2} and fed to the subsequent layers to produces the stylized output feature E_i.

Discussion: Previous AST methods rely on heavy backbones to extract style information from style images at the cost of high computational overhead and long inference time. In contrast, we encode local geometric structures and global properties from style images into compact representations (16-dimension). Our method significantly reduces the computational complexity and produces notable inference speedup. Besides, in contrast to heavy style extractors, the pluggable style representation contains sufficient style information in a very small storage space. It quickly expands to model parameters, which greatly reduces the model size and storage burden.

3.3 Framework Training

Training Loss The overall loss function consists of a content term, a style term, a reconstruction term, and a geometric term, which is defined as follows:

$$\mathcal{L} = \lambda_c \mathcal{L}_c(I_i, c) + \lambda_s \mathcal{L}_s(I_i, s_i) + \lambda_{ae} \mathcal{L}_{ae}(I_0, c) + \lambda_{geo} \mathcal{L}_{geo}(c, i), \tag{4}$$

where λ_c, λ_s, λ_{ae} and λ_{geo} are set to 1,10, 0.01 and 0.01, respectively. We show the training process in Algorithm 1.

(1) Content loss and style loss:

Similar to previous works [7,8,54], we define content and style loss as follows:

$$\mathcal{L}_c(I_i, c) = \sum_{l \in \{l_c\}} ||VGG^l(I_i) - VGG^l(c)||_2, \tag{5}$$

$$\begin{aligned} \mathcal{L}_s(I_i, s_i) = \sum_{l \in \{l_s\}} (&||\mu(VGG^l(I_i)) - \mu(VGG^l(s_i))||_2 + \\ &||\sigma(VGG^l(I_i)) - \sigma(VGG^l(s_i))||_2), \end{aligned} \tag{6}$$

where VGG^l refers to features extracted from the l-th layer in a pre-trained VGG-16 [35]. $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and variance of extracted features, respectively.

(2) Reconstruction loss:

In order to allow users to edit stylization degree, a style representation f_0 is set to represent the content and style information of the content image itself (*i.e.*, a set of auto-encoder parameters is stored in f_0). The reconstruction loss is as follows:

$$\mathcal{L}_{ae}(I_0, c) = ||I_0 - c||_2. \tag{7}$$

(3) Geometric loss:

Geometric consistency, which is introduced in recent works [11, 30], reduces the space of possible translation to preserve the scene geometry. Inspired by geometric consistency, we implement geometric consistency loss to promote stylized results.

$$\begin{aligned}\mathcal{L}_{geo}(c, i) = \sum_{t \in \{T\}} (&||\text{SaMST}(c, f_i) - t^{-1}(\text{SaMST}(t(c), f_i))||_1 \\ &+||\text{SaMST}(t(c), f_i) - t(\text{SaMST}(c, f_i))||_1),\end{aligned} \tag{8}$$

where $\{T\}$ represent 8 distinct patterns of flip and rotation.

Algorithm 1: Pipeline to train a general model

Data : content images, N style images $\{s_1, s_2, ..., s_N\}$ and corresponding style indices $\{1, 2, ..., N\}$.
Target: SCB $= \{f_0, f_1, f_2, ..., f_N\}$, SaMST.
Initial : $f \leftarrow \mathbb{1}$ for every f in SCB, $iterations \leftarrow$ training iterations, $r \leftarrow 0$.

while $r \leq iterations$ **do**
- Randomly sample b content images $C = \{c_j\}$ and b style indices $Y = \{y_j\}$ ($j \in \{1, ..., b\}$ and $y_j \in \{1, ..., N\}$) as one mini-batch. According to Y, select corresponding style images $S = \{s_{y_j}\}$ and style representations $F = \{f_{y_j}\}$
- Inference: stylized images $I = \text{SaMST}(C, F)$ (*i.e.*, $I = \{c_j s_{y_j}\} = \{I_{y_j}\}$); reconstructed content images $I_0 = \text{SaMST}(C, f_0)$
- Loss: $\mathcal{L} = \lambda_c \mathcal{L}_c(I, C) + \lambda_s \mathcal{L}_s(I, S) + \lambda_{ae} \mathcal{L}_{ae}(I_0, C) + \lambda_{geo} \mathcal{L}_{geo}(C, Y)$
- update: SaMST and (F, f_0)
- $r \leftarrow r + 1$;

end

Incremental Training Previous MST methods need to finetune their models when extending new styles [55], which results in style catastrophic forgetting. In addition, the model size of these methods increases as the number of styles increases. For example, Stylebank [3] stores styles in heavy network structures and requires $1.18M$ parameters for each new style (as shown in Table 1).

To address problems above, we propose a novel scheme for style extension. SaMST is trained on a large number of style images, and we believe that it already has strong generalization to address wider style domain. When adding new styles, we fix the SaMST and just update new style representations iteratively. We show the incremental training process in Algorithm 2.

Discussion: Our SaMST differs from previous MST methods [3, 10] in two aspects. (1) Architecture: Style-specific filters in Stylebank consist of vanilla convolution and instance norm layers ($1.18M$ parameters). In contrast, we store styles in much more

Algorithm 2: Incremental training

Data : content images, M incremental style images $\{s_{N+1}, s_{N+2}, ..., s_{N+M}\}$ and corresponding style indices $\{N+1, N+2, ..., N+M\}$.
Fixed : Previous stable SCB: $\text{SCB}_{fixed} = \{f_0, f_1, ...f_N\}$, SaMST.
Target: $\text{SCB}_{add} = \{f_{N+1}, f_{N+2}, ..., f_{N+M}\}$.
Initial : $f \leftarrow \mathbb{1}$ for every f in SCB_{add}, $iterations \leftarrow$ incremental training iterations, $r \leftarrow 0$.

while $r \leq iterations$ **do**
- Randomly sample b content images $C = \{c_j\}$ and b style indices $Y = \{y_j\}$ ($j \in \{1, ..., b\}$ and $y_j \in \{N+1, ..., N+M\}$) as one mini-batch. Then select corresponding style images $S = \{s_{y_j}\}$ and style representations $F = \{f_{y_j}\}$
- Inference: stylized images $I = \text{SaMST}(C, F)$ (*i.e.*, $I = \{c_j s_{y_j}\} = \{I_{y_j}\}$).
- Loss: $\mathcal{L} = \lambda_c \mathcal{L}_c(I, C) + \lambda_s \mathcal{L}_s(I, S) + \lambda_{geo} \mathcal{L}_{geo}(C, Y)$
- update: F
- $r \leftarrow r + 1$;

end
$\text{SCB} = \text{SCB}_{fixed} + \text{SCB}_{add}$

compact representations (16 parameters). To utilize these representations, we design a universal and novel SaMST with SConv, SRAdaIN and SCM. (2) Loss: As compared to widely used losses, we introduce an additional geometric loss to promote generation quality.

4 Experiments

4.1 Implementation Details

We use MS-COCO [29] as content dataset and select $50k$ style images from WikiArt [31] and the Internet. Style representation length C in Sec. 3.1 is set to 16. And generator contains 3 SABs. During training, content images are rescaled to 256×256 pixels and style images are rescaled to 512×512 pixels. 8 content-style image patch pairs are randomly selected as a mini-batch. We adopt the Adam optimizer [25] to train the whole network for $3M$ iterations. The initial learning rate is set to 0.001 and decreased to half every $0.75M$ iterations. During test phase, SaMST can handle any input size as it is fully convolutional. More implementation details are available in supplemental material.

4.2 Comparison with Prior Arts

We compare our SaMST to recent state-of-the-art ST methods, including CDST [41], AdaConv [2], StyTR2 [8], EFDM [56], CAPVST [47], ATK [59], MicroAST [46], S2WAT [54] and Stylebank [3]. Among them, Stylebank[3] is a typical MST method while others are AST methods. We obtain the results of the methods by following their

[3] Stylebank [3] contains $1.18M$ parameters for each style, which results in heavy storage burden. So we just train stylebank on 500 style images.

Fig. 4: Visualization results of image details produced by different methods on a 2K image from Flickr2K dataset. The whole content image is shown in Fig. 2.

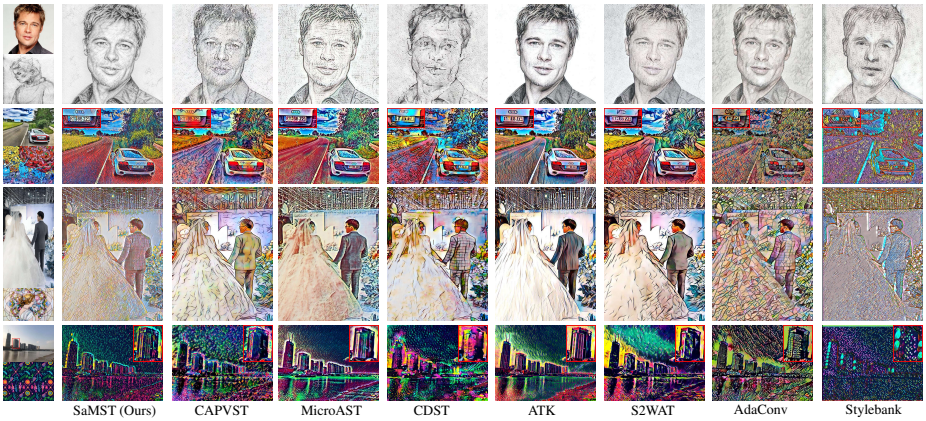

Fig. 5: Qualitative comparison with the state of the art. Please zoom in for best view.

official code with default configurations. More comparison results with more methods are available in supplemental material.

Qualitative Comparison We show the visual comparisons in Fig. 4 and Fig. 5. In Fig. 4, MicroAST [46] and CDST [41], which are also designed as lightweight and fast models, suffer from severe performance drop. MicroAST produces compression artifacts (*e.g.*, the 4^{th} row), while CDST breaks image local geometric structures and content details (*e.g.*, the 1^{st} row). AdaConv [2] is proposed to capture more local structures from style images. As shown in the 1^{st} row, AdaConv focuses on local geometry of style images excessively, resulting in severe image distortion and artifacts. CAPVST [47], ATK [59] and S2WAT [54] are also hard to achieve satisfied results. As for MST method, Stylebank [3] transfers insufficient style properties and destroys content details. In contrast, our SaMST produces more clear details and achieves higher perceptual quality (*e.g.*, the texts in the 1^{st} row and license plate in the 3^{rd} row). And our SaMST captures sufficient colors and textures from style images, while it also keeps content structures accurately (*e.g.*, the 2^{nd} row and the 4^{th} row).

As shown in Fig. 5, our SaMST achieves best visual quality to keep balance between style and content in the whole image. For example, our method captures detailed content information rather than other methods (*e.g.*, 2^{nd} and 4^{th} row). Moreover, stylization generated by our SaMST are more closed to style images. In contrast, other methods

Table 1: Quantitative comparison of the style transfer methods.Methods marked with * are MST approaches, while other methods are AST approaches. The **best** and second best results are highlighted, respectively. Run time and FLOPs are evaluated on 512×512 images. "+" represents that the method expands new styles without forgetting. 'OIP' is short for 'once inference parameters', which refers to the number of parameters involved in one stylization inference for a certain style.

Metric	CDST [41]	AdaConv [2]	StyTR2 [8]	EFDM [56]	CAPVST [47]	ATK [59]	MicroAST [46]	S2WAT [54]	Stylebank* [3]	SaMST* (Ours)
ArtFID [48] ↓	57.02	52.31	46.78	58.10	53.97	34.87	59.46	38.74	64.02	**25.20**
CF [45] ↑	0.320	0.363	0.475	0.351	0.397	0.515	0.335	0.452	0.296	**0.538**
GE +LP [45] ↑	1.125	1.482	1.337	1.295	1.184	1.265	1.312	1.388	1.089	**1.515**
FLOPs (G) ↓	39.52	145.68	1283.45	63.30	179.89	291.44	11.06	582.62	35.48	**5.31**
Time (ms) ↓	247.67	15.54	194.76	14.92	33.36	24.01	3.96	94.88	5.60	**1.03**
Params (M) ↓	2.42	62.83	35.39	7.01	4.09	11.18	**0.47**	64.96	590.79	0.91
OIP (M) ↓	2.42	62.83	35.39	7.01	4.09	11.18	0.47	64.96	1.97	**0.11**
Style Capacity ↑	∞	∞	∞	∞	∞	∞	∞	∞	500+	50k+

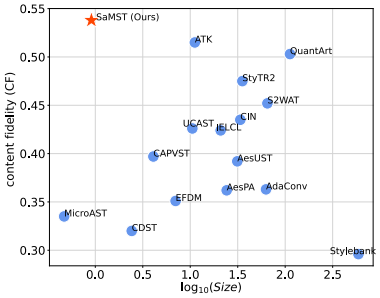

Fig. 6: Comparison of CF score [45] and model size (M).

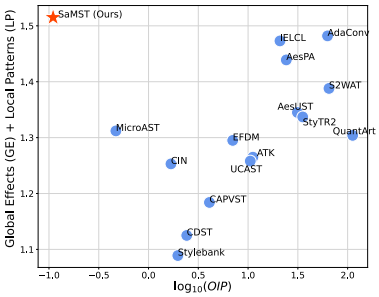

Fig. 7: Comparison of GE+LP score [45] and OIP (M).

suffer from content distortion (*e.g.*, AdaConv) and insufficient stylization (*e.g.*, Stylebank), and so on.

Quantitative Comparison We resort to some quantitative metrics to better evaluate the proposed method. The results are shown in Table 1, Fig. 1, Fig. 6 and Fig. 7.
(1) Stylization Quality: Wright *et al.* [48] propose ArtFID to measure stylized image quality from both style and content. Wang *et al.* [45] propose content fidelity (CF), global effects (GE) and local patterns (LP) to evaluate the quality of ST. In detail, CF measures the faithfulness to content characteristics; GE assesses the stylization quality in terms of the global effects like global colors and holistic textures; LP assesses the stylization quality in terms of the similarity and diversity of the local style patterns. We collect 500 content images and 100 style images to synthesize $50k$ stylized images for each method and show their average metric scores in Table 1. Our SaMST achieves best results on the 3 metrics, indicating that it can transfer sufficient style patterns while better preserving the content details.
(2) Efficiency and Application: As shown in Table 1 and Fig. 1, our SaMST achieves the best stylization results with the lowest computation quantity and inference time. This is because our style representation scheme contains enough style information compared with heavy VGG extractor. As for OIP (*i.e.*, number of parameters for each style) in Table 1, our SaMST requires only $0.11M$ parameters while other methods suffer from parameter redundancy (*e.g.*, $64.96M$ in S2WAT [54]). This indicates that our SaMST is easy to be deployed on edge devices or cloud servers, which has strong application value. Moreover, when adding new styles, size of Stylebank increases quickly

Table 2: Ablation Study. Note that, we use CIN [10] with 3 resblocks as baseline for fair comparison (*i.e.*, model 1). All model variants are trained on $50k$ style images. Runtime and FLOPs are evaluated on 512×512 images.

Method	Param (M)	OIP (M)	Time (ms)	FLOPs (G)	Sconv	SRAdaIN	SCM	$\mathcal{L}_{geo}$	Metric		
									ArtFID↓	CF↑	GE+LP↑
Model 1	54.58	0.18	4.08	7.16	✖	✖	✖	✖	45.83	0.417	1.191
Model 2	55.30	0.10	0.76	5.35	✔	✖	✖	✖	38.62	0.382	1.306
Model 3	0.99	0.19	4.32	7.12	✖	✔	✖	✖	47.89	**0.594**	1.214
Model 4	0.91	0.11	0.98	5.31	✔	✔	✖	✖	33.24	0.467	1.408
Model 5	0.91	0.11	1.03	5.31	✔	✔	✔	✖	28.36	0.486	1.464
Model 6 (Ours)	0.91	0.11	1.03	5.31	✔	✔	✔	✔	**25.20**	0.528	**1.515**

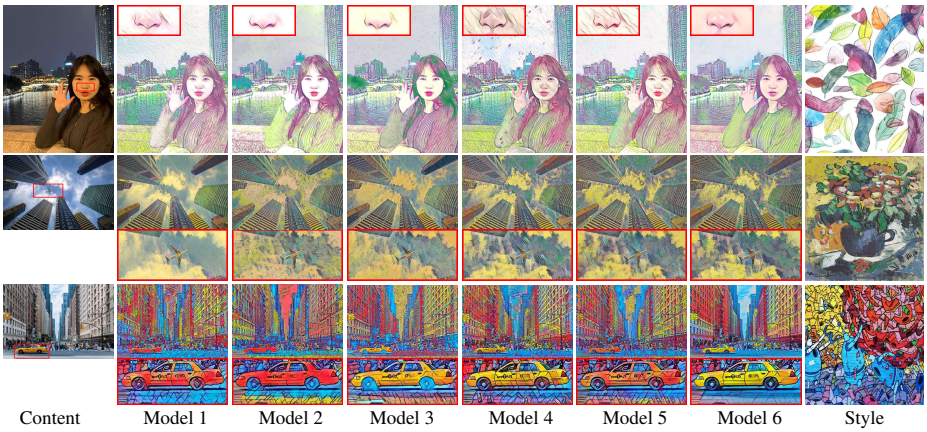

Fig. 8: Ablation study of SaMST. We also display the zoomed patch in the red box for better evaluation. The settings of the model variants are shown in Table 2.

($1.18M$ per style), resulting in heavy storage burden. In contrast, size of SaMST increases slowly (16 per style). It is because that SCB in SaMST is capable of storing a large number of image styles in limited storage space. Then our SaMST can cover the need for various image styles in application scenarios.

4.3 Model Analysis

We demonstrate the effectiveness of our proposed components in this section. And more model analysis can be found in supplemental material.

(1) SConv: We propose SConv to reproduce local geometric structures from style to content. To validate its effectiveness, we introduce a model variant (model 1 in Table 2) by removing all designs. In generator, we replace the common convolution layer by our SConv layer to obtain model 2. Although model 2 achieves better inference time, model size increases more quickly. This is because both style representations in SCB and learnable variables in conditional IN layer grow with the number of styles. Moreover, SConv helps achieve significantly higher ArtFID and GE+LP score than model 1. From the second scene in Fig. 8, model 2 produces image textures that closed to the style image.

(2) SRAdaIN: To make stylized images capture global properties from style images, SRAdaIN is employed in our method. To demonstrate its effectiveness, we add SRAdaIN to obtain model 1 to obtain model 3 for comparison. It can be observed from Table 2

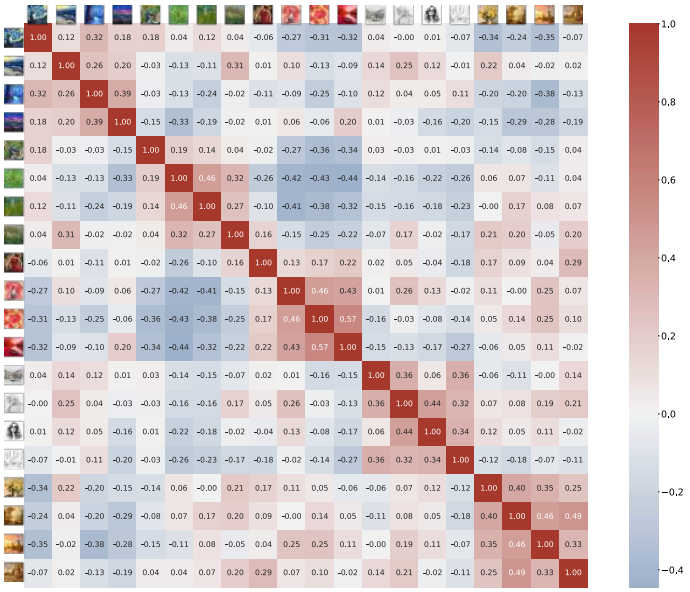

Fig. 9: Style correlation matrix.

that the SRAdaIN improves CF and GE+LP score, which indicates that style representations contains more accurate global style properties. Moreover, the model size is much smaller than model 1. In addition, we further develop model 4 by adding SConv to model 3. With both SConv and SRAdaIN, model 4 produces significant higher metrics as compared to model 1. From the first scene in Fig. 8, model 4 produces soft tones similar with the style image. Moreover, from the third scene in Fig. 8, model 4 produces sharper edges of higher perceptual quality.

(3) SCM: Although model 4 produces soft tones, it also produces artifacts on stylized images (*e.g.*, the second scene in Fig. 8). As we introduce SCM to model 4, we obtain model 5 to achieve better visual quality. And model 5 achieves better CF score, which indicates that stylized images suffer from less image distortion.

(4) Geometric Consistency Loss: The geometric consistency loss is introduced to preserve scene geometry. The loss $\mathcal{L}_{geo}$ is employed in our method. We add $\mathcal{L}_{geo}$ to model 5 to obtain model 6 for comparison. Model 6 achieves best ArtFID and GE+LP score and second best CF score. It demonstrates that model 6 achieves the better balance between style transformation and content preservation. As shown in the first and third scene of Fig. 8, model 6 produces clearer stylized results and image details. In contrast, models trained without $\mathcal{L}_{geo}$ produce messy stylized image details.

(5) Style Correlation: Our method contains style information in compact representations. The representations are correlated with the visualization of style images. To demonstrate this, we randomly select 20 styles and corresponding style representations. We calculate correlation matrix of the selected styles. Then we visualize the result in Fig. 9. The correlation matrix shows that the visually similar styles have higher correlation coefficient (*e.g.*, high correlation coefficient between the sketch-wise styles). The correlation matrix provides strong evidence that the style representations distinguish styles in semantic and visual similarity.

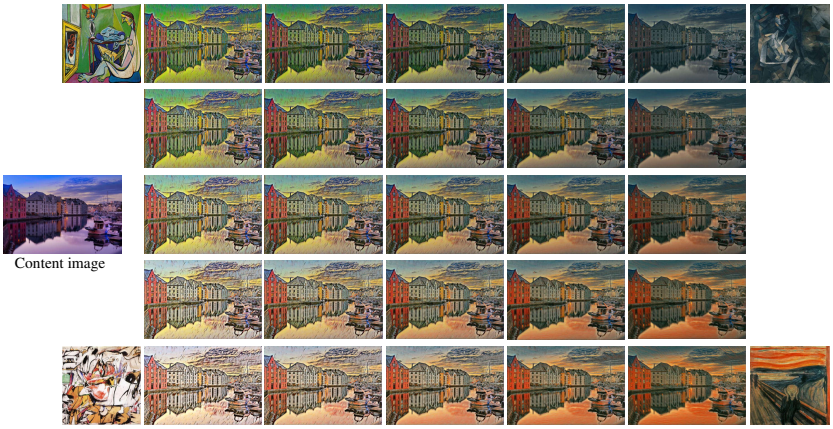

Fig. 10: Visualization results by style interpolation.

Fig. 11: Visualization results by style interpolation. We also show the conversion between photo-realistic images and artistic images.

4.4 User Interaction

(1) Style Interpolation: To blend a set of K styles $s_1, s_2, ..., s_K$, we interpolate between style representations $f_1, f_2, ..., f_K$ to obtain overall style representation $f_{mixed} = \sum_{i=1}^{K} w_i f_i$ such that $\sum_{i=1}^{K} w_i = 1$. Then the stylized image is generated:

$$I_{mixed} = \text{SaMST}(c, f_{mixed}). \tag{9}$$

The results are shown in Fig. 10 and Fig. 11. Figure 10 illustrates the results interpolated with the different weight levels. The style pattern gradually shift to other style patterns by changing the interpolation weights. As shown in Fig. 11, our SaMST can converts between photorealistic images and artistic images (*e.g.*, photorealistic content to artistic styles in 1^{st} row and artistic content to photorealistic styles in 2^{nd} row).

The degree of style transfer can be controlled during training by adjusting the style weight λ_s in Eq. 4. Our SaMST allows an alternative to implement content-style trade-off:

$$I_{\alpha i} = \text{SaMST}(c, (1-\alpha) f_0 + \alpha f_i), \tag{10}$$

where α is the degree factor to control the stylized images (as shown in Fig. 12).
(2) Incremental Training: We enable an incremental training for new styles, which has comparable learning time to the online-learning method [12], while model size grows slower than previous method [3]. Specifically, we fix model 6 in Table 2, then train the

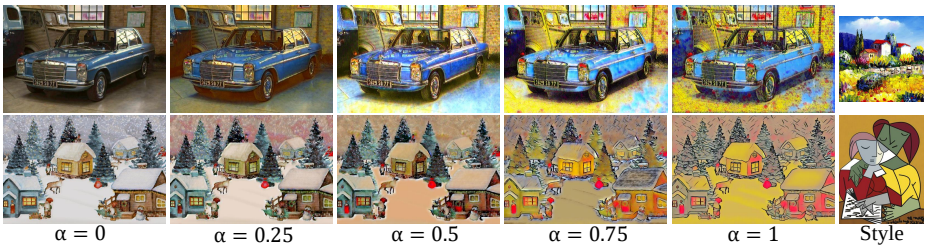

Fig. 12: Content-style trade-off. We can control the balance between content and style by changing the factor α in Eq. 10.

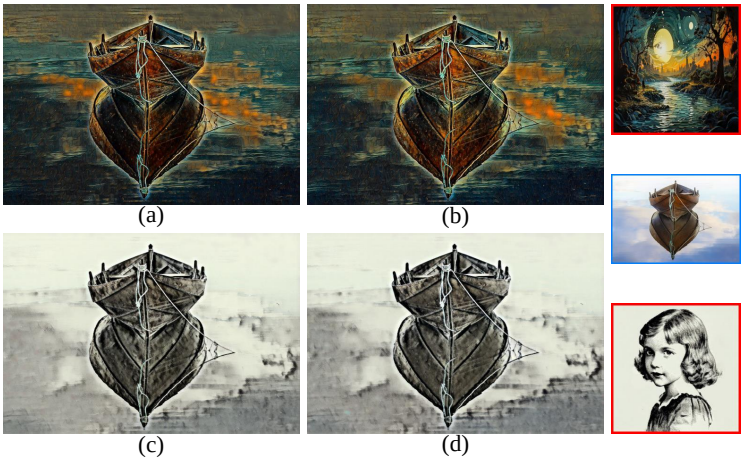

Fig. 13: Comparison between incremental training (a)(c) and fresh training (b)(d). The content and styles are shown on the right.

style representations for new styles (as shown in Algorithm 2). The style converges very fast since only the 16-dimension style representation would be updated in iterations instead of the whole model. In our experiments, when training with NVIDIA RTX 3090 GPU and given training image size of 256, it only takes around 1 minute with about $3k$ iterations to train a new style. Figure 13 shows stylized results of new styles by incremental training. Compared to fresh training (*i.e.*, retraining the whole network with the new styles), our learning scheme obtain very similar stylized results. More incremental training results can be found in supplemental material.

5 Conclusion

In this paper, we propose a style representation learning scheme to store accurate style information. Moreover, we introduce a lightweight and practical style-aware multi-style transfer (SaMST) network to achieves efficient ST. In addition, we propose a incremental training scheme to expand new styles without forgetting. It is demonstrated that our style representation learning scheme can extract accurate and robust style information. Experimental results show that our network achieves state-of-the-art performance for ST task.

Acknowledgement. This work was partially supported by the Guangdong Basic and Applied Basic Research Foundation (2022B1515020103), and the Shenzhen Science and Technology Program (No. RCYX20200714114641140).

References

1. Champandard, A.J.: Semantic style transfer and turning two-bit doodles into fine artworks. arXiv preprint arXiv:1603.01768 (2016) 4
2. Chandran, P., Zoss, G., Gotardo, P., Gross, M., Bradley, D.: Adaptive convolutions for structure-aware style transfer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7972–7981 (2021) 1, 3, 4, 5, 8, 9, 10
3. Chen, D., Yuan, L., Liao, J., Yu, N., Hua, G.: Stylebank: An explicit representation for neural image style transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1897–1906 (2017) 1, 2, 3, 4, 7, 8, 9, 10, 13
4. Chen, H., Wang, Z., Zhang, H., Zuo, Z., Li, A., Xing, W., Lu, D., et al.: Artistic style transfer with internal-external learning and contrastive learning. Advances in Neural Information Processing Systems **34**, 26561–26573 (2021) 3
5. Chiu, T.Y., Gurari, D.: Pca-based knowledge distillation towards lightweight and content-style balanced photorealistic style transfer models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7844–7853 (2022) 4
6. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258 (2017) 4
7. Deng, Y., Tang, F., Dong, W., Huang, H., Ma, C., Xu, C.: Arbitrary video style transfer via multi-channel correlation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 1210–1217 (2021) 6
8. Deng, Y., Tang, F., Dong, W., Ma, C., Pan, X., Wang, L., Xu, C.: Stytr2: Image style transfer with transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11326–11336 (2022) 3, 6, 8, 10
9. Deng, Y., Tang, F., Dong, W., Sun, W., Huang, F., Xu, C.: Arbitrary style transfer via multi-adaptation network. In: Proceedings of the 28th ACM international conference on multimedia. pp. 2719–2727 (2020) 1, 3
10. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. arXiv preprint arXiv:1610.07629 (2016) 1, 2, 3, 4, 5, 7, 11
11. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Zhang, K., Tao, D.: Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2427–2436 (2019) 7
12. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015) 1, 3, 13
13. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016) 1, 3
14. Gatys, L.A., Ecker, A.S., Bethge, M., Hertzmann, A., Shechtman, E.: Controlling perceptual factors in neural style transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3985–3993 (2017) 4
15. He, J., Dong, C., Qiao, Y.: Interactive multi-dimension modulation with dynamic controllable residual learning for image restoration. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16. pp. 53–68. Springer (2020) 6
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 4
17. Hong, K., Jeon, S., Lee, J., Ahn, N., Kim, K., Lee, P., Kim, D., Uh, Y., Byun, H.: Aespa-net: Aesthetic pattern-aware style transfer networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22758–22767 (2023) 1, 3

18. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017) 4
19. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE international conference on computer vision. pp. 1501–1510 (2017) 1, 3, 5
20. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. pmlr (2015) 4
21. Jin, J., Dundar, A., Culurciello, E.: Flattened convolutional neural networks for feedforward acceleration. arXiv preprint arXiv:1412.5474 (2014) 4
22. Jing, Y., Liu, X., Ding, Y., Wang, X., Ding, E., Song, M., Wen, S.: Dynamic instance normalization for arbitrary style transfer. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 4369–4376 (2020) 1, 2, 3, 4
23. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. pp. 694–711. Springer (2016) 1, 3, 4
24. Kaiser, L., Gomez, A.N., Chollet, F.: Depthwise separable convolutions for neural machine translation. arXiv preprint arXiv:1706.03059 (2017) 4
25. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 8
26. Kolkin, N., Salavon, J., Shakhnarovich, G.: Style transfer by relaxed optimal transport and self-similarity. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10051–10060 (2019) 4
27. Li, X., Liu, S., Kautz, J., Yang, M.H.: Learning linear transformations for fast image and video style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3809–3817 (2019) 1, 3
28. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Diversified texture synthesis with feed-forward networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3920–3928 (2017) 4
29. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014) 8
30. Maeda, S.: Unpaired image super-resolution using pseudo-supervision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 291–300 (2020) 7
31. Phillips, F., Mackintosh, B.: Wiki art gallery, inc.: A case for critical thinking. Issues in Accounting Education **26**(3), 593–608 (2011) 8
32. Shen, F., Yan, S., Zeng, G.: Neural style transfer via meta networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8061–8069 (2018) 2, 4
33. Sheng, L., Lin, Z., Shao, J., Wang, X.: Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8242–8250 (2018) 1, 3
34. Sifre, L., Mallat, S.: Rotation, scaling and deformation invariant scattering for texture discrimination. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1233–1240 (2013) 4
35. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 3, 6

36. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015) 4
37. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.: Texture networks: Feed-forward synthesis of textures and stylized images. arXiv preprint arXiv:1603.03417 (2016) 1, 3
38. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016) 1, 3
39. Ulyanov, D., Vedaldi, A., Lempitsky, V.: 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. pp. 6924–6932 (2017) 4
40. Vanhoucke, V.: Learning visual representations at scale. ICLR invited talk **1**(2) (2014) 4
41. Wang, H., Li, Y., Wang, Y., Hu, H., Yang, M.H.: Collaborative distillation for ultra-resolution universal style transfer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1860–1869 (2020) 4, 8, 9, 10
42. Wang, L., Wang, Y., Dong, X., Xu, Q., Yang, J., An, W., Guo, Y.: Unsupervised degradation representation learning for blind super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10581–10590 (2021) 4
43. Wang, M., Liu, B., Foroosh, H.: Factorized convolutional neural networks. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 545–553 (2017) 4
44. Wang, Z., Zhao, L., Chen, H., Qiu, L., Mo, Q., Lin, S., Xing, W., Lu, D.: Diversified arbitrary style transfer via deep feature perturbation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7789–7798 (2020) 4
45. Wang, Z., Zhao, L., Chen, H., Zuo, Z., Li, A., Xing, W., Lu, D.: Evaluate and improve the quality of neural style transfer. Computer Vision and Image Understanding **207**, 103203 (2021) 4, 10
46. Wang, Z., Zhao, L., Zuo, Z., Li, A., Chen, H., Xing, W., Lu, D.: Microast: Towards super-fast ultra-resolution arbitrary style transfer. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 2742–2750 (2023) 2, 3, 4, 8, 9, 10
47. Wen, L., Gao, C., Zou, C.: Cap-vstnet: content affinity preserved versatile style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18300–18309 (2023) 8, 9, 10
48. Wright, M., Ommer, B.: Artfid: Quantitative evaluation of neural style transfer. In: DAGM German Conference on Pattern Recognition. pp. 560–576. Springer (2022) 2, 4, 10
49. Wu, X., Hu, Z., Sheng, L., Xu, D.: Styleformer: Real-time arbitrary style transfer via parametric style composition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14618–14627 (2021) 3
50. Xia, B., Zhang, Y., Wang, Y., Tian, Y., Yang, W., Timofte, R., Van Gool, L.: Knowledge distillation based degradation estimation for blind super-resolution. arXiv preprint arXiv:2211.16928 (2022) 4
51. Xu, H., Li, Q., Zhang, W., Zheng, W.: Styleremix: An interpretable representation for neural image style transfer. arXiv preprint arXiv:1902.10425 (2019) 4
52. Yanai, K., Tanno, R.: Conditional fast style transfer network. In: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval. pp. 434–437 (2017) 2
53. Yeh, M.C., Tang, S., Bhattad, A., Zou, C., Forsyth, D.: Improving style transfer with calibrated metrics. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 3160–3168 (2020) 4
54. Zhang, C., Xu, X., Wang, L., Dai, Z., Yang, J.: S2wat: Image style transfer via hierarchical vision transformer using strips window attention. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 7024–7032 (2024) 1, 3, 6, 8, 9, 10

55. Zhang, H., Dana, K.: Multi-style generative network for real-time transfer. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. pp. 0–0 (2018) 1, 2, 3, 4, 7
56. Zhang, Y., Li, M., Li, R., Jia, K., Zhang, L.: Exact feature distribution matching for arbitrary style transfer and domain generalization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8035–8045 (2022) 1, 3, 8, 10
57. Zhang, Y., Tang, F., Dong, W., Huang, H., Ma, C., Lee, T.Y., Xu, C.: Domain enhanced arbitrary image style transfer via contrastive learning. In: ACM SIGGRAPH 2022 conference proceedings. pp. 1–8 (2022) 3
58. Zhang, Y., Tang, F., Dong, W., Huang, H., Ma, C., Lee, T.Y., Xu, C.: A unified arbitrary style transfer framework via adaptive contrastive learning. ACM Transactions on Graphics **42**(5), 1–16 (2023) 3
59. Zhu, M., He, X., Wang, N., Wang, X., Gao, X.: All-to-key attention for arbitrary style transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23109–23119 (2023) 1, 3, 8, 9, 10