Mamba-ST: State Space Model for Efficient Style Transfer

Filippo Botti Alex Ergasti Leonardo Rossi Tomaso Fontanini
Claudio Ferrari Massimo Bertozzi Andrea Prati
University of Parma, Department of Engineering and Architecture
Parma, Italy
{filippo.botti, alex.ergasti, leonardo.rossi, claudio.ferrari2,
massimo.bertozzi, andrea.prati}@unipr.it

Abstract

The goal of style transfer is, given a content image and a style source, generating a new image preserving the content but with the artistic representation of the style source. Most of the state-of-the-art architectures use transformers or diffusion-based models to perform this task, despite the heavy computational burden that they require. In particular, transformers use self- and cross-attention layers which have large memory footprint, while diffusion models require high inference time. To overcome the above, this paper explores a novel design of Mamba, an emergent State-Space Model (SSM), called Mamba-ST, to perform style transfer. To do so, we adapt Mamba linear equation to simulate the behavior of cross-attention layers, which are able to combine two separate embeddings into a single output, but drastically reducing memory usage and time complexity. We modified the Mamba's inner equations so to accept inputs from, and combine, two separate data streams. To the best of our knowledge, this is the first attempt to adapt the equations of SSMs to a vision task like style transfer without requiring any other module like cross-attention or custom normalization layers. An extensive set of experiments demonstrates the superiority and efficiency of our method in performing style transfer compared to transformers and diffusion models. Results show improved quality in terms of both ArtFID and FID metrics. Code is available at https://github.com/FilippoBotti/MambaST.

1. Introduction

Style Transfer is a deep learning technique aiming to generate a new image which has the content (*e.g* objects, layout) of a given image (*i.e.* the content image) and the style (*e.g.* color or texture structure) of another image (*i.e.* the style image). Style Transfer has been largely studied [6, 11, 24, 25, 27, 29, 33, 42] and there exist several models

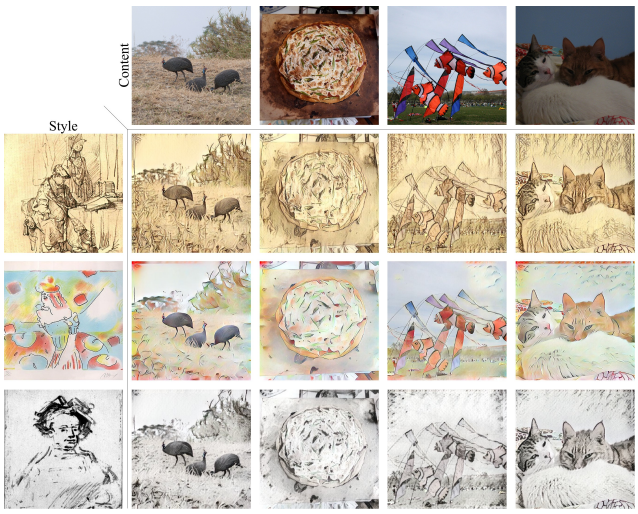

Figure 1. Examples of generated images from our Mamba model given a style and a content image

which can perform it with good results in terms of quality and consistency between content and style. Recently, style transfer was also extended to text-based models by substituting the style image with a textual description [25, 27]. In this work, we will not tackle or compare with this category of models as they largely differ from standard style transfer architectures. General approaches follow the encoder-decoder pipeline and perform style transfer in an intermediate layer between these two sections [24, 29, 33, 42]. However, these models struggle to find a relation between content and style, leading to poor quality results in terms of image details. For this reason, recent architectures (*e.g.* [11]) leverage the capability of transformers by taking advantage of cross-attention mechanism to improve the quality of the results. However, transformer-based architectures have a large memory requirement. On the other hand, several approaches [6, 43, 49] tried to leverage diffusion models as backbones for style transfer, but did not reach the same

quality as transformer-based architecture or, as stated in [6], required high inference time compared to the former.

On the contrary, State Space Models (SSM) [18,19] have recently shown results comparable to transformers for long sequence modeling and in vision task [20,23,50]. In particular, among these, Mamba [17] proved to be a competitive alternative to transformers, but with a much lower memory requirement. Additionally, Mamba complexity scales linearly with the sequence length, rather than quadratically, resulting in a fast inference especially if compared with diffusion models. Mamba was initially designed to work with 1D sequences (like words in a sentence), but was recently adapted to work with 2D vision data thanks to VMamba [32]. Recently, in a text style transfer architecture described by Wang *et al*. [41], Mamba building blocks demonstrated performance comparable to transformer-based architectures. However, additional Adaptive Layer Normalization (AdaLN) was necessary to fuse the style and content.

In this work, we propose a way to adapt the inner equations of Mamba to perform style transfer (Fig. 1) with a novel architecture called Mamba-ST (Mamba Style Transfer). The main part of Mamba-ST is a novel block, called Mamba-ST Decoder (MSTD), which is able to fuse the style information extracted from an image with the content of a different image. More in detail, both content and style are modelled as a sequence of patch embeddings and are fed to our proposed MSTD. In order to perform the fusion, we modified the Mamba internal matrices to mimic the functionality of a cross-attention layer, yet maintaining the core properties of SSMs. By doing so, our solution enables the interaction between style and content image without the need of additional modules, like Adaptive Layer Normalization (AdaLN). Our contributions can be summarized as:

- We designed a cross-attention-like method inside SSMs. This is done by adapting the mathematical formulation of internal matrices so that additional layers like AdaLN are not required, whilst maintaining the same properties of the basic Mamba block.
- A novel vision-based Mamba architecture, called Mamba-ST, which is able to perform style transfer with comparable results with respect to transformer- and diffusion-based architectures.
- The proposed approach allows a better memory usage w.r.t. transformers and a much faster inference time compared with diffusion models.

2. Related Work

Style Transfer. The problem of style transfer has been widely studied in the literature [2, 4, 12, 22, 24, 28, 31]. The first attempt to transfer style was the one proposed by Gatys *et al*. [15], which shows that is possible to merge content and style features extracted by a CNN by solving an optimization problem. Later, the introduction of AdaIN [24] allowed to perform arbitrary style transfer adjusting the mean and the variance of the content image and align them with the ones of the style image. Thanks to its efficiency, AdaIN became a very popular architecture. Later, the advent of transformers [31,34] showed how self-attention mechanism can improve the quality of the results by finding stronger relations between the style and the content. Subsequently, Deng *et al*. [11] introduced a fully-transformer-based architecture, StyTr^2, which, combined with a new content-aware positional encoding scheme, outperforms state-of-the-art methods for style transfer. Despite the capability reached in terms of quality, these methods heavily depend on transformers, so they scale quadratically with the image size, which limits their use only on small images.

Recently, diffusion-based style transfer methods [6, 43, 49] showed how to leverage the generative capability of diffusion models in order to perform style transfer. InST [49] captured the information of the style with a text-based inversion method and then transfer it. StyleDiffusion [43] introduced a CLIP-based style disentanglement loss to disentangle style and content in the CLIP image space. Despite the quality of the images produced with these architectures, the content and the style are not perfectly merged together and the results are not yet on par with the ones generated by transformer-based models. Finally, StyleID [6] proposed a new style transfer method which exploits the knowledge of Stable Diffusion 1.4 [35] without requiring supervision or optimisation. StyleID simply substitutes key and value of content with those of styles inside self-attention layers, but, despite the improved results w.r.t. previous methods, it still requires high inference time compared to the one of transformer-based architectures. In order to solve the mentioned problems regarding memory usage and speed, while maintaining quality and coherency during style transfer, we propose a new full Mamba-based architecture.

State Space Models. Despite the well known superiority of transformer-based architecture in vision tasks [3, 14, 37, 46, 47], one of the most crucial problems of these architecture remains their quadratic complexity and high memory requirement. For this reason, recently, several works tried to overcome this issue [5,7,8,13,40]. In particular, State Space Models (SSMs) [18, 19] were inspired by control systems theory and have been recently introduced in deep learning field in order to take the advantage of their linear complexity [23,32,39,50]. Structured State Space Models [18] proposed a new parametrization for SSM in order to get the advantage of parallelization during training and achieve high speed during inference. Mamba [17] was recently introduced as an improved SSM. Its main contribution consists in making the SSM parameters input-dependent. Since its

superiority compared to other SSMs in terms of memory usage, time complexity and quality of the results, Mamba has been widely applied in deep learning, from NLP [39], to vision field [32, 50] like super-resolution [20, 36] or even diffusion models [23]. Recently, a first attempt of using Mamba for text-driven style transfer was presented [41], but it failed to take the advantage of the inner SSMs matrix and still used custom normalization layers like AdaLN for transferring the style. In this context, Mamba was utilized not for merging content and style directly, but primarily as a feature extractor to exploit its fast inference speed. Notably, the combined AdaLN+Mamba model exhibited limited generalization capabilities, requiring separate training for each new (text, image) pair. Consequently, Mamba facilitated more rapid interaction between image patches, while the style fusion process was primarily handled by the AdaLN layers. On the contrary, we adapt the inner equation of Mamba and provide a full Mamba-based architecture which is able to merge content and style without any other module like AdaLN or cross-attention. Moreover, our method learns how to transfer style and, once trained, can be used with any image-style pair without the need to be retrained.

3. Method

In this section, the proposed architecture is introduced by firstly describing Mamba and then showing how its inner equation can be adapted to perform style transfer.

3.1. Background on Mamba

State-Space Models, like S4 [18], learn to map a 1-D sequence $x(t) \in \mathbb{R}$ to another 1-D sequence $y(t) \in \mathbb{R}$ as output, maintaining an internal state space $h(t) \in \mathbb{R}^N$, where N is the state size. Differently from other sequence learning approaches like transformers, SSMs scale linearly w.r.t. the sequence length. SSMs are described by a linear ordinary differential equation (ODE) system:

$$\begin{aligned} h'(t) &= Ah(t) + Bx(t) \\ y(t) &= Ch(t) + Dx(t) \end{aligned} \tag{1}$$

where the matrices $A \in \mathbb{R}^{N\times N}, B \in \mathbb{R}^{N\times 1}, C \in \mathbb{R}^{1\times N}$ and $D \in \mathbb{R}$ are learnable. In order to be usable in deep learning architectures, a discretization phase is applied to the system. Here we decided to use the zero-order holder (ZOH) rule, where Δ represents the step parameter. Specifically, by denoting with $\bar{A}, \bar{B}$ the discretized matrices, written in a RNN form, the equation becomes:

$$\begin{aligned} h_k &= \bar{A}h_{k-1} + \bar{B}x_k \\ y_k &= Ch_k + Dx_k \end{aligned} \tag{2}$$

Given their ability to compress the context in a finite state, recurrent networks are more efficient than transformers, yet their main limitation becomes how well the state can compress the context information [17] (*e.g.* understanding the most relevant words in a sentence while ignoring the others). For this reasons, Gu *et al*. introduced Mamba [17], by adding an input dependency on the matrices B, C and Δ:

$$B = \mathrm{Lin_B}(x), C = \mathrm{Lin_C}(x), \Delta = \mathrm{Lin_\Delta}(x) \tag{3}$$

with Lin_* being a linear, fully-connected layer.

This contribution, combined with an efficient selective scan algorithm for temporal coherency, maintains all the computation efficiency but with a state that is able to better memorize and understand the context information.

3.2. Overall Architecture

Given the ability of Mamba to better understand contextual information while maintaining the efficiency of a recurrent network, in this work we aim to adapt Mamba to perform style transfer. To this end, we propose Mamba-ST, whose architecture is shown in Figure 2 (a). Our system is composed of three main components: *(i)* two Mamba Encoders that encode content and style images, respectively, *(ii)* a Mamba Style Transfer (ST) Decoder (MSTD) that fuses together content and style information and *(iii)* a CNN decoder to rearrange the decoder output back to an image.

Both content and style images are divided into patches and projected into 1D embeddings using a PatchEmbed layer [44]. The PatchEmbed takes the images $I \in \mathbb{R}^{C\times H\times W}$ as input, and produces a series of embeddings $t \in \mathbb{R}^{D\times(h\cdot w)}$, where $h = \frac{H}{p}$ and $w = \frac{W}{p}$ (with p the patch size), and D is the hidden dimension of each embedding. Then, we employ the two different domain-encoders, which take as input the corresponding embedding set (*i.e.* content and style), to learn the visual representations of the images. After that, content and style representations extracted by the Mamba encoders are fed to the Mamba-ST Decoder (see Fig. 2 (c)) which is tasked to merge the two streams of information. Finally, a depatchify block is used to obtain a feature map of size $D \times h \times w$ that the CNN decoder transforms to obtain the output image of size $C \times H \times W$.

3.3. Mamba Encoder

Each of the encoders is composed of three Mamba Encoder layers, illustrated in Fig. 2 (b), whose structure is derived from VMamba [32], except for a skip connection that we added between each layer, in order to avoid vanishing gradient problem. After an initial layer normalization, the embeddings are fed to the Base Visual SSM (Base VSSM), illustrated in Fig. 2 (d). The Base VSSM structure is achieved by substituting the S6 module employed by Mamba to perform the selective scan, with its 2D counterpart called 2D-SSM (Fig 2 (e)).

The 2D-SSM learns the matrices A, B, C and Δ and models the state of the layer and the output, which is the

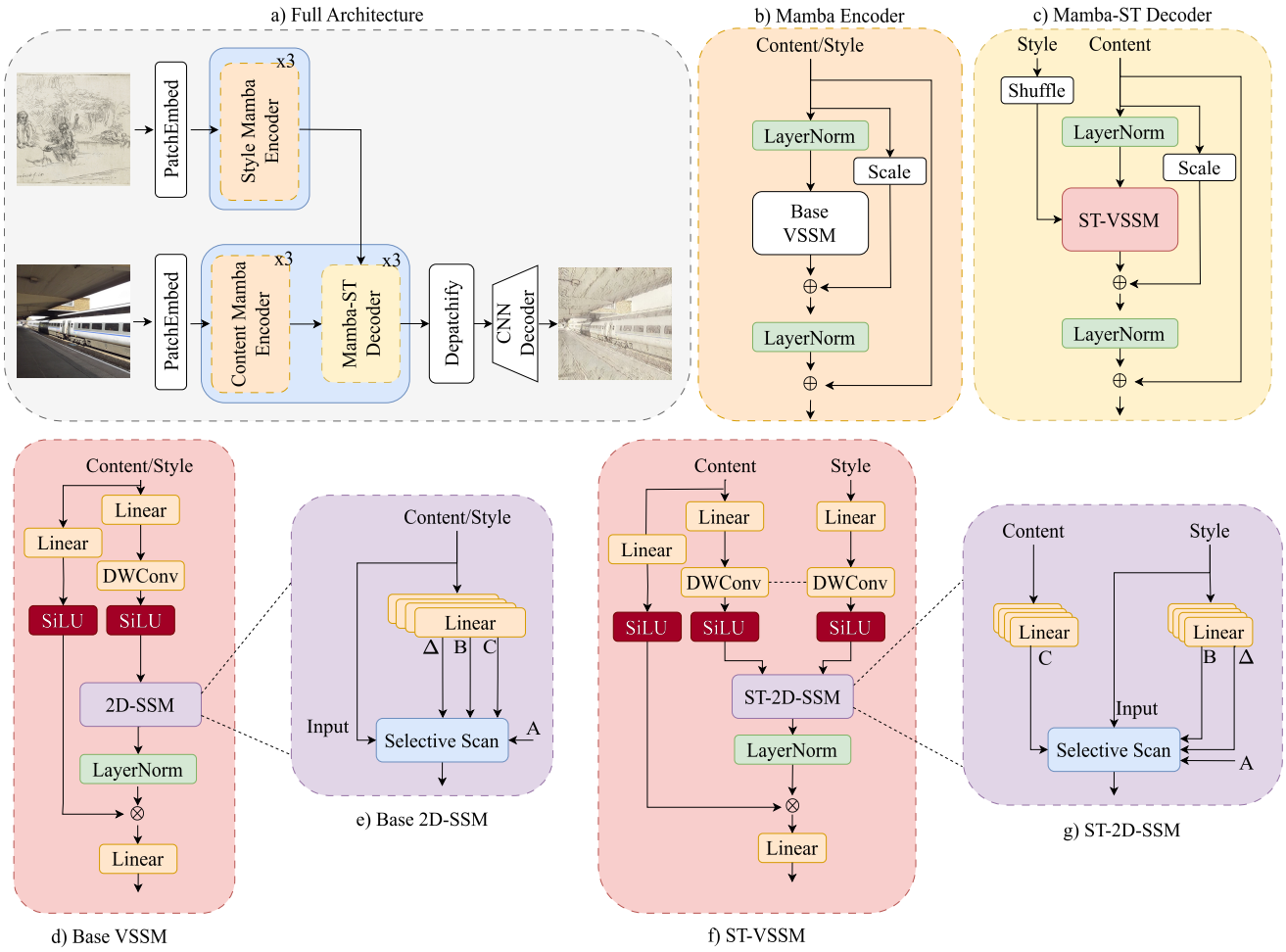

Figure 2. *a*) Mamba-ST full architecture. It takes as input a content and a style image and generates the content image stylized as the style image. *b*) Mamba encoder derived from [32] with an additional skip connection (rightmost). *c*) Our Mamba-ST Decoder, which takes both style and content as input. In particular, style embeddings are shuffled before passing to ST-VSSM in order to loose spatial information, maintaining only higher level information. *d*) The inner architecture of the Base VSSM. *e*) The inner architecture of the Base 2D-SSM. *f*) Our ST-VSSM. Notably, DWConv is shared among content and style embedding. *g*) Our modified ST 2D-SSM, where the matrices A,B and Δ are computed from the style, the input of the selective scan are the style embedding and the matrix C is calculated using the content.

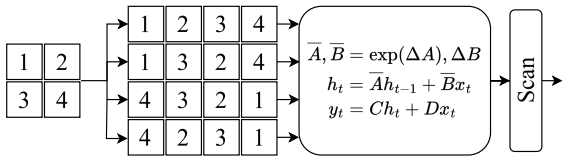

Figure 3. The 2D selective scan with a 2×2 example image.

encoded visual representation of the image. The core of 2D-SSM is the selective scan mechanism [17] adapted for 2D sequences (Fig. 3). We follow [32] and use four different scan directions to maintain spatial information. Then, for each scan, we calculate state and output following Eq. (2) and merge them together by reordering and summing them. The algorithm is presented in Alg. 1 in the supplementary.

3.4. Mamba-ST Decoder

The Mamba-ST Decoder (MSTD) is tasked to merge the content and the style visual representations, extracted by the two Mamba encoder blocks, in a single representation, effectively performing style transfer. Its overall structure is similar to the encoder one, but differs in two main characteristics. First, it takes as input both style and content embeddings, which are fed to a modified version of the base VSSM called ST-VSSM (see Fig. 2 (f)). Secondly, inside the ST-VSSM, the 2D-SSM is replaced by the proposed ST-2D-SSM (see Fig. 2 (g)) which is specifically designed to fuse content and style information.

Recently, [1] and [9] showed a duality between Mamba equation and transformer self-attention. In particular,

they suggested that query Q, key K and value V matrices employed in the self attention equation ($Att = \text{Softmax}(\frac{Q \cdot K^T}{\sqrt{d}}) \cdot V$) can be expressed as:

$$Q \approx C, K \approx B, V \approx X \tag{4}$$

where X is the input sequence. Following this symmetry between Mamba and self-attention, our intuition was to mimic a cross-attention mechanism by letting A, B and Δ matrices be dependent from the style source s, while making C dependent from the content x as follows:

$$B = \text{Lin}_\text{B}(s), \Delta = \text{Lin}_\Delta(s), C = \text{Lin}_\text{C}(x) \tag{5}$$

Similarly, we decided to pass the encoded style features instead of the content as input sequence for the ST-2D-SSM. More in detail, based on Eq. (2), we incorporate the style information inside the state, as the equation for the state depends only on A and B matrices:

$$h_k = \bar{A} h_{k-1} + \bar{B} s_k \tag{6}$$

Then, the output is made dependent on both style and content since C is derived from the content image:

$$y_k = C h_k \tag{7}$$

This allows to effectively merge content and style information and to perform style transfer.

The inner selective-scan mechanism inside the decoder layer is the same as the one inside the encoder. Furthermore, in order to remove content details from the style that could jeopardize the style transfer, we decided to apply a random shuffle to the style embedding, as shown in Fig. 2 (c). In this way, the hidden state of the model loses every information about the content of the style picture, leaving only style information. In Alg. 2 in the supplementary materials, we provide the full algorithm description of the decoder block.

3.5. Losses

We train our model using two perceptual losses. The content loss $\mathcal{L}_C$ focuses on preserving the content of the original image, while the style loss $\mathcal{L}_S$ aims to transfer the style of the source image to the target image.

We implement the two losses using a pretrained VGG19 model following [2, 11, 24]. Let x_c be the content image, x_s be the style image and x_g be the generated image. Given N_l the number of the layers selected from the VGG19, we define $\phi_i(x)$ as the features extracted from the i-th layer with as input the image x. The content loss is defined as:

$$\mathcal{L}_C = \frac{1}{N_l} \sum_{i=0}^{N_l} \left\| \phi_i\left(x_g\right) - \phi_i\left(x_c\right) \right\|_2 \tag{8}$$

The style loss is instead defined as:

$$\begin{aligned} \mathcal{L}_S = \frac{1}{N_l} \sum_{i=0}^{N_l} (&\left\| \mu\left(\phi_i\left(x_g\right)\right) - \mu\left(\phi_i\left(x_s\right)\right) \right\|_2 \\ &+ \left\| \sigma\left(\phi_i\left(x_g\right)\right) - \sigma\left(\phi_i\left(x_s\right)\right) \right\|_2) \end{aligned} \tag{9}$$

where $\mu(\cdot)$ is the mean of a given feature map and $\sigma(\cdot)$ is the standard deviation of a given feature map.

Furthermore, we also use two identity losses [11, 34]. These help in learning better representations for both content and style. Let x_g^c be the image generated using the content image x_c as both style and content information, and x_g^s be the generated image using the style image x_s as both style and content information, we then define:

$$\mathcal{L}_{id1} = ||x_g^c - x_c||_2 + ||x_g^s - x_s||_2, \tag{10}$$

$$\begin{aligned} \mathcal{L}_{id2} = \frac{1}{N_l} \sum_{i=0}^{N_l} (&||\phi(x_g^c) - \phi(x_c)||_2 \\ &+ ||\phi(x_g^s) - \phi(x_s)||_2) \end{aligned} \tag{11}$$

The final loss which we use to train our model is:

$$\mathcal{L} = \lambda_C \mathcal{L}_C + \lambda_S \mathcal{L}_S + \lambda_{id1} \mathcal{L}_{id1} + \lambda_{id2} \mathcal{L}_{id2}, \tag{12}$$

with $\lambda_C = 7, \lambda_S = 10, \lambda_{id1} = 70$ and $\lambda_{id2} = 1$ in order to balance the magnitude of each loss [11].

4. Experiments

Implementation details We use the COCO dataset [30] as our content dataset and the WikiArt dataset [38] as our style dataset to train our model. We adopt the same hyperparameters setting as [11], with the exception of the learning rate, which we set to 0.00005 without utilizing any warm-up period. Our model is trained for a total of 160,000 iterations with a batch size of 8 on a single NVIDIA L40S GPU. Moreover, we set p (*i.e.* the patch size) to 8.

Evaluation details We compare our model both qualitatively and quantitatively with several state-of-the-art models designed for image-to-image style transfer: StyleID [6], AesPA-Net [22], StyTr^2 [11], AdaAttn [31] AdaIN [24]. We intentionally avoid comparing with style transfer methods which use textual descriptions as style condition instead of an image (including StyleMamba[1] [41]) since it would not be a fair comparison.

To quantitatively evaluate our model, we employ four primary metrics: ArtFID [45], FID [21], LPIPS [48], and CFSD [6]. We did not consider content loss $\mathcal{L}_C$ and style loss $\mathcal{L}_S$ as complementary evaluation metrics since [6] noted that utilizing these losses for both training and evaluation would introduce evaluation biases.

[1] Furthermore, no code is available for this paper

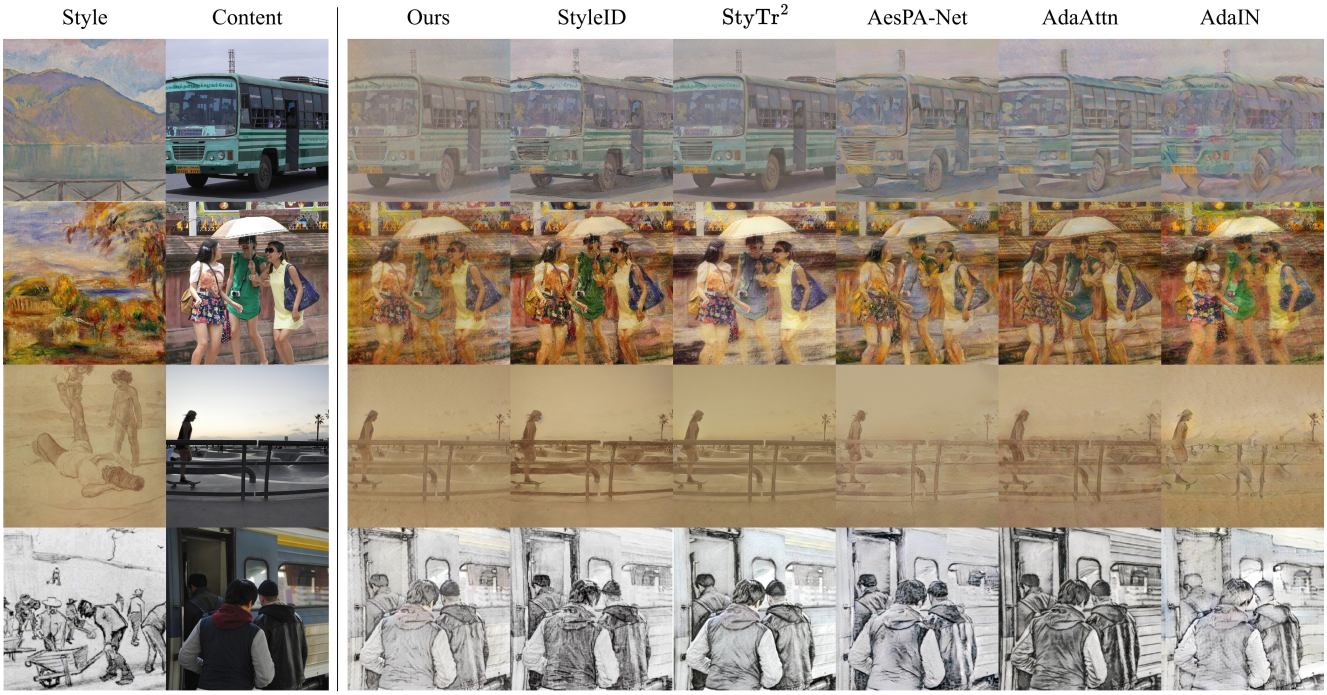

Figure 4. Visual comparison with the current state-of-the-art models.

More in detail, FID [21] measures the overall similarity between generated and style images distributions. LPIPS [48] evaluates the content preservation between a source image and the stylized image, thereby measuring how well the content is preserved in the stylized image. ArtFID [45] is a metric found to be highly correlated with human judgment, and fuses together style transfer and content preservation. ArtFID is calculated as $(1 + \text{FID}) \cdot (1 + \text{LPIPS})$.

CFSD (Content Feature Structural Distance) [6] is a metric designed to address the limitations of LPIPS [6, 16]. Indeed, LPIPS utilizes a feature space extracted by a pre-trained AlexNet [26], which is trained on the ImageNet dataset [10] for classification tasks. However, ImageNet is known to be texture-biased [16], meaning that style injection can influence the LPIPS measure, potentially leading to inaccurate assessments of content preservation. CFSD has been hence introduced as distance metric based on the spatial correlation between image patches.

For each model, we calculate the metrics on 800 generated samples obtained by randomly sampling 40 style images and 20 content images following [6]. Furthermore, we calculate the inference time (s) to generate the 800 images and memory usage (MebiByte, MiB) with batch size 1.

Quantitative analysis In Tab. 1 quantitative evaluation of several state-of-the-art models w.r.t. the proposed system is presented. Notably, our method outperforms previous architecture in ArtFID, which, as previously discussed, is strongly correlated with human judgment. Additionally, our method achieves the lowest FID, indicating superior style transfer to the content image. On the other side, in terms of content preservation metrics (*i.e.* LPIPS and CFSD) our model has comparable, albeit slight worse performance. This trade-off highlights our model capability to transfer the style at the cost of a marginal reduction of content preservation. However, we argue that, compared to the best SOTA model (AdaIN) in terms of FID and ArtFID we are able to also improve LPIPS and CFSD and, if compared with the best SOTA model (StyleID) in terms of LPIPS or CFSD, we are able to improve both FID and ArtFID.

AdaIN, thanks to its lightweight backbone, is also the most efficient model in terms of both time and memory usage. Specifically, it can generate 800 images (derived from 20 content images combined with 40 style images) in just 12.26 seconds, while maintaining minimal memory consumption. At the same time though, even if it surprisingly achieves the second best results in ArtFID and FID, it heavily falls behind (it is indeed the worst) in terms of LPIPS and CFSD meaning that it fails in preserving the correct content in the samples. The second fastest model is AdaAttn; however, it demands significantly higher memory capacity. Finally, our proposed model strikes a balance between time and memory usage, requiring low memory while delivering an acceptable inference time. This is particularly evident

	Metrics				Time and Memory usage	
Model	ArtFID ↓	FID ↓	LPIPS ↓	CFSD ↓	Time (s) ↓	Memory Usage (MiB) ↓
AdaIN [24]	27.81	16.80	0.56	0.35	12.26	824
AdaAttN [31]	30.81	19.46	0.51	0.32	23.89	5554
StyTr^2 [11]	29.31	18.77	**0.48**	0.32	52.35	2160
AesPA-Net [22]	35.45	22.85	0.49	0.33	165.99	4184
StyleID [6]	28.65	18.29	0.49	**0.29**	2744.38	19930
Ours	**27.11**	**16.75**	0.53	0.33	24.70	1414

Table 1. Performance comparison of the SOTA models and our proposed model on 512×512 resolution. The best result for each metric is highlighted in bold, while the second-best result is marked in red. Time is calculated generating the entire 800 stylized images. Memory usage is calculated with batch 1.

	Metrics				Time and Memory usage	
	ArtFID ↓	FID ↓	LPIPS ↓	CFSD ↓	Time (s) ↓	Memory Usage (MiB) ↓
Content as input 2D-SSM	28.17	17.76	0.50	**0.33**	24.71	1414
# Scan Directions = 1	34.69	21.49	0.54	0.63	15.84	1288
# Scan Directions = 2	28.17	17.91	**0.49**	**0.33**	19.05	1294
Dim state=8	27.26	16.79	0.53	**0.33**	22.93	1424
Dim state=32	28.34	17.33	0.55	0.34	27.73	1446
w/o random in inference	27.63	**16.25**	0.60	0.36	24.39	1430
Ours	**27.11**	16.75	0.53	**0.33**	24.70	1414

Table 2. Various ablation studies. The best result for each metric is highlighted in bold, while the second-best result is marked in red. Time is calculated generating the entire 800 stylized images. Memory usage is calculated with batch 1.

when comparing with diffusion-based models like StyleID, which is both memory-intensive and time-consuming.

Qualitative analysis Qualitative comparisons are shown in Fig. 4. As it can be seen, we are able to achieve comparable results w.r.t. the current state-of-the-art models. Looking at the figure, AdaIN is able to correctly apply the style, but the overall content is greatly altered as LPIPS and CFSD value in Tab. 1 already showed. On the other side, StyTr^2, AesPA-Net and AdaAttn sometimes struggle to maintain color coherency when applying the style. For example, in the second row the middle girl's dress is turned to blue instead of green which was the color in the original content image. Finally, StyleID is able to produce coherent images that present both the original content and the style features, but with very high contrast and saturation in the colors which is typical of diffusion models. This may alter the faithful application of styles characterized by soft colors (see first and third rows) or, on the other side, push too much the application of high-contrast styles (see second row). The fourth row instead provides examples where every method is capable of producing satisfactory results. Finally, our method represents a trade-off between the previous style transfer models. We are able to both apply styles while maintaining the correct color coherency of the content image, but, at the same time, without excessively changing the saturation and contrast of the generated samples.

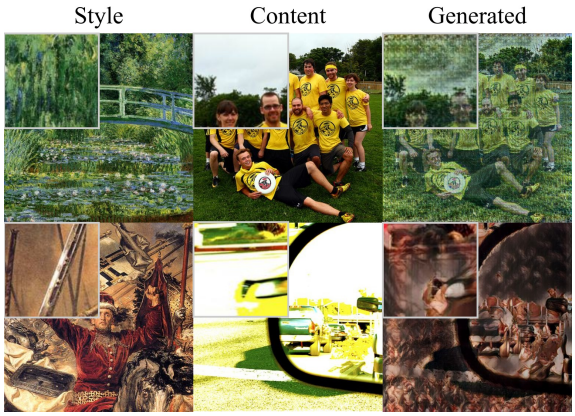

Figure 5. Zoomed results which show the patch problem inside the results. Gaps are present between each patch in the results and the model failed to uniformly apply the style.

Despite the excellent results, sometimes the model fails to correctly apply the style, as shown in Fig. 5. Sometimes it reproduces non-homogeneous patches inside the output images with a gap between them. A possible reason is that Mamba-ST inherits RNNs limitations; in some cases, it is difficult to ensure continuity between the patches due to the memorization of context information inside the state.

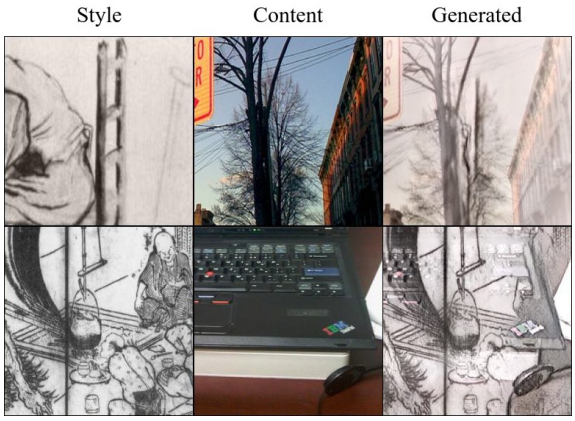

Figure 6. Ablation study of our model without the random shuffle in the style embedding. The generated images become a blend of both context and style images.

Ablation Finally, we performed several ablation studies to determine the optimal configuration for our system. Results are shown in Tab. 2. First, we investigated the effect of passing the content image instead of style to the selective scan. This resulted in overall worse performance. Also looking at the mathematical analysis provided in the supplementary materials, we ultimately opted for passing the style as input to the selective scan. The second and third rows of Tab. 2 show the performance of our model when trained using 1 or 2 scan directions instead of 4. When adopting a single direction, the overall performance drop largely while 2 directions improve a bit, yet still being lower than the final architecture with 4 directions. The inference time with 4 scans is only slightly worse. In the fourth and fifth rows of Tab. 2 we show the effect of varying the dimension of the Mamba internal state. The performance improves when increasing the dimension from 8 to 16 (our final model), but drop when further increasing it from 16 to 32. Based on these findings, we select 16 as the dimension of the internal state. Finally, in the sixth row of the table, we tested removing the shuffling module at inference. The improved FID indicates that the model more accurately captures the stylistic features of the reference image, leading to good stylized images. Nevertheless, this adjustment led to a large deterioration in LPIPS. This outcome suggests that the model might overly emphasize the style at the expense of preserving the content fidelity. Without shuffling, the spatial and structural information within the style image is maintained, ultimately leading to a blended version of the content and style. The shuffling module is thus necessary even at inference for effectively capturing high-level style information.

In conclusion, Fig. 6 presents examples generated by our best model when shuffling is disabled during the training phase too. The absence of shuffling prevents the model from isolating style information in the generated output. Instead, it retains content details from the style image as well. Consequently, the output is a blended version of both the content and style images, rather than a proper transfer of style.

5. Conclusion

In this work we investigated the adaptation of Mamba inner equation for image driven style transfer, leveraging its lightweight capability to lower time consumption and memory usage w.r.t. the state of the art. Specifically, we propose a new Mamba block, called Mamba-ST Decoder which is able to accept two streams of information as input and fuse them together in a single output. Finally, we provide an extensive set of comparison with SOTA models and a comprehensive set of ablation studies proving the efficacy of the proposed solution.

6. Acknowledgments

This work was partially funded by "Partenariato FAIR (Future Artificial Intelligence Research) - PE00000013, CUP J33C22002830006" funded by the European Union - NextGenerationEU through the italian MUR within NRRP.

References

[1] Ameen Ali, Itamar Zimerman, and Lior Wolf. The hidden attention of mamba models. *arXiv preprint arXiv:2403.01590*, 2024. 4

[2] Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. Artflow: Unbiased image style transfer via reversible neural flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 862–871, 2021. 2, 5

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[4] Prashanth Chandran, Gaspard Zoss, Paulo Gotardo, Markus Gross, and Derek Bradley. Adaptive convolutions for structure-aware style transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7972–7981, 2021. 2

[5] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. 2

[6] Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8795–8805, 2024. 1, 2, 5, 6, 7

[7] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023. 2

[8] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022. 2

[9] Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024. 4

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[11] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11326–11336, 2022. 1, 2, 5, 7

[12] Yingying Deng, Fan Tang, Weiming Dong, Wen Sun, Feiyue Huang, and Changsheng Xu. Arbitrary style transfer via multi-adaptation network. In *Proceedings of the 28th ACM international conference on multimedia*, pages 2719–2727, 2020. 2

[13] Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. Longnet: Scaling transformers to 1,000,000,000 tokens. *arXiv preprint arXiv:2307.02486*, 2023. 2

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 2

[15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 2

[16] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 6

[17] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 2, 3, 4

[18] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *The International Conference on Learning Representations (ICLR)*, 2022. 2, 3

[19] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021. 2

[20] Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. Mambair: A simple baseline for image restoration with state-space model. *arXiv preprint arXiv:2402.15648*, 2024. 2, 3

[21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5, 6

[22] Kibeom Hong, Seogkyu Jeon, Junsoo Lee, Namhyuk Ahn, Kunhee Kim, Pilhyeon Lee, Daesik Kim, Youngjung Uh, and Hyeran Byun. Aespa-net: Aesthetic pattern-aware style transfer networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22758–22767, 2023. 2, 5, 7

[23] Vincent Tao Hu, Stefan Andreas Baumann, Ming Gui, Olga Grebenkova, Pingchuan Ma, Johannes Fischer, and Bjorn Ommer. Zigma: Zigzag mamba diffusion model. *arXiv preprint arXiv:2403.13802*, 2024. 2, 3

[24] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017. 1, 2, 5, 7

[25] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2426–2435, 2022. 1

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 6

[27] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18062–18071, 2022. 1

[28] Gihyun Kwon and Jong Chul Ye. Diffusion-based image translation using disentangled style and content representation. *arXiv preprint arXiv:2209.15264*, 2022. 2

[29] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30, 2017. 1

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[31] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6649–6658, 2021. 2, 5, 7

[32] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model, 2024. 2, 3, 4

[33] Ming Lu, Hao Zhao, Anbang Yao, Yurong Chen, Feng Xu, and Li Zhang. A closed-form solution to universal style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5952–5961, 2019. 1

[34] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5880–5888, 2019. 2, 5

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[36] Yuan Shi, Bin Xia, Xiaoyu Jin, Xing Wang, Tianyu Zhao, Xin Xia, Xuefeng Xiao, and Wenming Yang. Vmambair: Visual state space model for image restoration. *arXiv preprint arXiv:2403.11423*, 2024. 3

[37] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7464–7473, 2019. 2

[38] Wei Ren Tan, Chee Seng Chan, Hernan E Aguirre, and Kiyoshi Tanaka. Improved artgan for conditional synthesis of natural image and artwork. *IEEE Transactions on Image Processing*, 28(1):394–409, 2018. 5

[39] Junxiong Wang, Tushaar Gangavarapu, Jing Nathan Yan, and Alexander M Rush. Mambabyte: Token-free selective state space model. *arXiv preprint arXiv:2401.13660*, 2024. 2, 3

[40] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 2

[41] Zijia Wang and Zhi-Song Liu. Stylemamba: State space model for efficient text-driven image style transfer. *arXiv preprint arXiv:2405.05027*, 2024. 2, 3, 5

[42] Zhizhong Wang, Lei Zhao, Haibo Chen, Lihong Qiu, Qihang Mo, Sihuan Lin, Wei Xing, and Dongming Lu. Diversified arbitrary style transfer via deep feature perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7789–7798, 2020. 1

[43] Zhizhong Wang, Lei Zhao, and Wei Xing. Stylediffusion: Controllable disentangled style transfer via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7677–7689, 2023. 1, 2

[44] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 3

[45] Matthias Wright and Björn Ommer. Artfid: Quantitative evaluation of neural style transfer. In *DAGM German Conference on Pattern Recognition*, pages 560–576. Springer, 2022. 5, 6

[46] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5791–5800, 2020. 2

[47] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10502–10511, 2019. 2

[48] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5, 6

[49] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10146–10156, 2023. 1, 2

[50] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024. 2, 3