–Supplementary Material–
SaMam: Style-aware State Space Model for Arbitrary Image Style Transfer

Hongda Liu[1], Longguang Wang[2], Ye Zhang[1], Ziru Yu[1], 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

1. Derivation Details of S7 Block

In this section we formalize S7 block, which efficiently generates stylised feature via global convolution.

1.1. Global Convolution by Vectorized Sequence

As shown in Eq. 3, the output can be computed by a RNN form. For the discretized version in our task, the parameters and input sequence are vectorized. The input sequence $\mathbf{x} \in \mathtt{L} * \mathtt{E}$. So Eq. 3 can be rewrite as follow:

$$\begin{aligned} \mathbf{h}_i &= \overline{\mathbf{A}}_i \mathbf{h}_{i-1} + \overline{\mathbf{B}}_i \mathbf{x}_i, \\ \mathbf{y}_i &= \mathbf{C}_i \mathbf{h}_i + \mathbf{D}\mathbf{x}_i, \end{aligned} \tag{I}$$

where $i \in [1, \mathtt{L}]$ represents time step and initial implicit latent state $\mathbf{h}_0 \in \mathtt{N} * \mathtt{E}$ is zero-initialized. And the i^{th} input token $\mathtt{x}_i \in 1 * \mathtt{E}$, the weight parameters $\overline{\mathbf{A}}_i \in \mathtt{N} * \mathtt{E}$, $\overline{\mathbf{B}}_i \in \mathtt{N} * \mathtt{E}$, $\mathbf{C}_i \in 1 * \mathtt{N}$ and $\mathbf{D} \in \mathtt{E}$. To simplify the calculus, we remove $\mathbf{D}\mathbf{x}_i$ since it can be seen as a skip connection multiplied by a scale factor $\mathbf{D}$. Then we can focus on i^{th} step output vector $\mathbf{y}_i$:

$$\begin{aligned} \mathbf{y}_i &= \mathbf{C}_i \mathbf{h}_i \\ &= \mathbf{C}_i(\overline{\mathbf{A}}_i \mathbf{h}_{i-1} + \overline{\mathbf{B}}_i \mathbf{x}_i) \\ &= \mathbf{C}_i(\overline{\mathbf{A}}_i(\overline{\mathbf{A}}_{i-1}\mathbf{h}_{i-2} + \overline{\mathbf{B}}_{i-1}\mathbf{x}_{i-1}) + \overline{\mathbf{B}}_i \mathbf{x}_i) \\ &\quad ..... \\ &= \mathbf{C}_i \overline{\mathbf{B}}_i \mathbf{x}_i + \mathbf{C}_i \overline{\mathbf{A}}_1 \overline{\mathbf{B}}_{i-1} \mathbf{x}_{i-1} + \mathbf{C}_i (\prod_{j=1}^{2} \overline{\mathbf{A}}_j) \overline{\mathbf{B}}_{i-2} \mathbf{x}_{i-2} + \\ &\quad ... + \mathbf{C}_i (\prod_{j=1}^{i-1} \overline{\mathbf{A}}_j) \overline{\mathbf{B}}_1 \mathbf{x}_1 + \mathbf{C}_i (\prod_{j=1}^{i} \overline{\mathbf{A}}_j) \mathbf{h}_0, \end{aligned} \tag{II}$$

where $\mathbf{h}_0 = \mathbf{0}$. Then the formulation of $\mathbf{y}_i \in 1 * \mathtt{E}$ can be simplified as:

$$\mathbf{y}_i = \mathbf{C}_i \sum_{m=1}^{i} (\mathbf{W}_{(m,i)} \mathbf{x}_m), \tag{III}$$

The weight $\mathbf{W}_{(m,i)} \in \mathtt{N} * \mathtt{E} (m \leq i \leq \mathtt{L})$ is represent as follow:

$$\mathbf{W}_{(m,i)} = (\prod_{n=1}^{i-m} \overline{\mathbf{A}}_n) \overline{\mathbf{B}}_m. \tag{IV}$$

The output sequence $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_\mathtt{L})^T \in \mathtt{L} * \mathtt{E}$. Then we can deduce the parallel convolution formulation of $\mathbf{y}$:

$$\begin{aligned} \mathbf{y} &= \Big(\mathbf{C}_1(\mathbf{W}_{(1,1)}\mathbf{x}_1), \mathbf{C}_2 \sum_{m=1}^{2} (\mathbf{W}_{(m,2)}\mathbf{x}_m), \mathbf{C}_3 \sum_{m=1}^{3} (\mathbf{W}_{(m,3)}\mathbf{x}_m), \\ &\quad ..., \mathbf{C}_\mathtt{L} \sum_{m=1}^{\mathtt{L}} (\mathbf{W}_{(m,\mathtt{L})}\mathbf{x}_m) \Big)^T \\ &= \Big((\mathbf{C}_1, \mathbf{C}_2, ..., \mathbf{C}_\mathtt{L}) \cdot \Big(\mathbf{W}_{(1,1)}\mathbf{x}_1, \sum_{m=1}^{2} (\mathbf{W}_{(m,2)}\mathbf{x}_m), ... \\ &\quad , \sum_{m=1}^{\mathtt{L}} (\mathbf{W}_{(m,\mathtt{L})}\mathbf{x}_m) \Big) \Big)^T \\ &= \Big(\mathbf{C}^T \cdot \Big(\mathbf{W}_{(1,1)}\mathbf{x}_1, \sum_{m=1}^{2} (\mathbf{W}_{(m,2)}\mathbf{x}_m), ..., \sum_{m=1}^{\mathtt{L}} (\mathbf{W}_{(m,\mathtt{L})}\mathbf{x}_m) \Big) \Big)^T \\ &= \mathbf{C} \cdot \Big((\mathbf{W}_{(1,1)}, \mathbf{W}_{(1,2)}, ..., \mathbf{W}_{(1,\mathtt{L})}) \circledast (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_\mathtt{L}) \Big)^T \\ &= \mathbf{C} \cdot \Big((\mathbf{W}_{(1,1)}, \mathbf{W}_{(1,2)}, ..., \mathbf{W}_{(1,\mathtt{L})}) \circledast \mathbf{x}^T \Big)^T \\ &= \mathbf{C} \cdot \Big((\mathbf{W}_{(1,1)}, \mathbf{W}_{(1,2)}, ..., \mathbf{W}_{(1,\mathtt{L})})^T \circledast \mathbf{x} \Big). \end{aligned} \tag{V}$$

Following Eq. V, we can get the structured global convolution kernel:

$$\overline{\mathbf{K}} = (\mathbf{W}_{(1,1)}, \mathbf{W}_{(1,2)}, ..., \mathbf{W}_{(1,\mathtt{L})})^T. \tag{VI}$$

Then we obtain the final formulation of the output sequence $\mathbf{y}$:

$$\mathbf{y} = \mathbf{C} \cdot (\overline{\mathbf{K}} \circledast \mathbf{x}), \tag{VII}$$

where $\circledast$ denotes convolution operation.

Following the protocol above, we can obtain model config. D in Table 2. When replacing S7 block by S6 block, the

*Corresponding author: Yulan Guo

Algorithm I SAVSSM (S6 Block) Process

Require: content feature $\mathbf{E_c}$: $(\mathtt{C}, \mathtt{H}, \mathtt{W})$,
style embedding $\mathbf{E_s}$: $(\mathtt{C}, \mathtt{H_s}, \mathtt{W_s})$
Ensure: stylized feature $\mathbf{E_{cs}}$: $(\mathtt{C}, \mathtt{H}, \mathtt{W})$
/* pre-proces content feature $\mathbf{E_c}$ */
$\mathbf{E'_c}$: $(\mathtt{C}, \mathtt{H}, \mathtt{W}) \leftarrow \mathbf{SAIN}(\mathbf{E_c}, \mathbf{E_s})$
$\mathbf{E'_c}$: $(\mathtt{E}, \mathtt{H}, \mathtt{W}) \leftarrow \mathbf{Linear}(\mathbf{E'_c})$
$\mathbf{E'_c}$: $(\mathtt{E}, \mathtt{H}, \mathtt{W}) \leftarrow \mathbf{SiLU}(\mathbf{SConv}(\mathbf{E'_c}, \mathbf{E_s}))$
/* process with four S6 Blocks, sequence length $\mathtt{L} = \mathtt{H} * \mathtt{W}$ */
for p in $\{path1, path2, path3, path4\}$ **do**
  $\mathbf{x}_p$: $(\mathtt{L}, \mathtt{E}) \leftarrow p(\mathbf{E'_c})$
  $\mathbf{B}_p$: $(\mathtt{L}, \mathtt{N}) \leftarrow \mathbf{Linear}^{\mathbf{B}}_p(\mathbf{x}_p)$
  $\mathbf{C}_p$: $(\mathtt{L}, \mathtt{N}) \leftarrow \mathbf{Linear}^{\mathbf{C}}_p(\mathbf{x}_p)$
  /* softplus ensures positive $\mathbf{\Delta}_p$ */
  $\mathbf{\Delta}_p$: $(\mathtt{L}, \mathtt{E}) \leftarrow \log(1 + \exp(\mathbf{Linear}^{\mathbf{\Delta}}_p(\mathbf{x}_p) + \mathbf{Parameter}^{\mathbf{\Delta}}_p))$
  /* parameters $\mathbf{A}$, $\mathbf{D}$ from concrete embedding space */
  $\mathbf{A}_p$: $(\mathtt{N}, \mathtt{E}) \leftarrow \mathbf{Parameter}^{\mathbf{A}}_p$
  $\mathbf{D}_p$: $(\mathtt{E},) \leftarrow \mathbf{Parameter}^{\mathbf{D}}_p$
  /* discretization process */
  $\overline{\mathbf{A}}_p$: $(\mathtt{L}, \mathtt{N}, \mathtt{E}) \leftarrow \exp(\mathbf{\Delta}_p \bigotimes \mathbf{A}_p)$
  $\overline{\mathbf{B}}_p$: $(\mathtt{L}, \mathtt{N}, \mathtt{E}) \leftarrow \mathbf{\Delta}_p \bigotimes \mathbf{B}_p$
  $\mathbf{y}_p$: $(\mathtt{L}, \mathtt{E}) \leftarrow \mathbf{SSM}(\overline{\mathbf{A}}_p, \overline{\mathbf{B}}_p, \mathbf{C}_p, \mathbf{D}_p)(\mathbf{x}_p)$
  $\mathbf{y}_p$: $(\mathtt{E}, \mathtt{H}, \mathtt{W}) \leftarrow \mathbf{Merge}(\mathbf{y}_p)$
end for
$\mathbf{E'_{cs}}$: $(\mathtt{E}, \mathtt{H}, \mathtt{W}) \leftarrow \mathbf{SAIN}(\mathbf{y}_{path1} + \mathbf{y}_{path2} + \mathbf{y}_{path3} + \mathbf{y}_{path4}, \mathbf{E_s})$
$\mathbf{E_{cs}}$: $(\mathtt{C}, \mathtt{H}, \mathtt{W}) \leftarrow \mathbf{Linear}(\mathbf{E'_{cs}}) + \mathbf{SCM}(\mathbf{E_c}, \mathbf{E_s})$
Return: $\mathbf{E_{cs}}$

SAVSSM process is shown in Algorithm I. Compared with config. A, parameters $\mathbf{A}$ and $\mathbf{D}$ are from concrete embedding space without introducing style information in hidden space updating.

1.2. Style-aware S6 Block (S7 Block)

As for each S7 block, the content image feature $\mathbf{E'_c} \in \mathtt{E}*\mathtt{H}*\mathtt{W}$ is firstly scanned to input sequence $\mathbf{x} \in \mathtt{L}*\mathtt{E}$ ($\mathtt{L} = \mathtt{H}*\mathtt{W}$). The timescale parameter $\mathbf{\Delta} \in \mathtt{L}*\mathtt{E}$ is input-dependent:

$$\mathbf{\Delta} = \mathbf{softplus}(\mathbf{Linear}^{\mathbf{\Delta}}(\mathbf{x})). \tag{VIII}$$

As for the weighting parameters, $\overline{\mathbf{B}} \in \mathtt{L}*\mathtt{N}*\mathtt{E}$ and $\mathbf{C} \in \mathtt{L}*\mathtt{N}$ are also predicted from input sequence $\mathbf{x}$:

$$\begin{aligned} \overline{\mathbf{B}} &= \mathbf{\Delta} \otimes \mathbf{Linear}^{\mathbf{B}}(\mathbf{x}), \\ \mathbf{C} &= \mathbf{Linear}^{\mathbf{C}}(\mathbf{x}), \end{aligned} \tag{IX}$$

where $\otimes$ refers to Einstein summation convention. Then different from S6 block, our style-aware weighting parameters $\overline{\mathbf{A}} \in \mathtt{L}*\mathtt{N}*\mathtt{E}$ and $\mathbf{D} \in \mathtt{E}$ are predicted from the style embedding $\mathbf{E_s} \in \mathtt{E}*\mathtt{H_s}*\mathtt{W_s}$:

$$\begin{aligned} \overline{\mathbf{A}} &= \exp\big(\mathbf{\Delta} \otimes \mathbf{Embedder}^{\mathbf{A}}(\mathbf{E_s})\big), \\ \mathbf{D} &= \mathbf{Embedder}^{\mathbf{D}}(\mathbf{E_s}). \end{aligned} \tag{X}$$

Table I. Quantitative comparison of the mamba-based ST methods. Run time and MACs are evaluated on 512×512 output resolution with a single NVIDIA RTX 3090 GPU.

Metrics	Mamba-based	
	Mamba-ST	SaMam (Ours)
LPIPS ↓	0.5058	0.3884
FID ↓	22.663	17.946
ArtFID ↓	35.632	26.305
CFSD ↓	0.4120	0.2703
MACs (G) ↓	859.9	77.1
Time (s) ↓	0.138	0.034
Params (M) ↓	40.03	18.50

Then we obtain the stylized output sequence $\mathbf{y} \in \mathtt{L}*\mathtt{E}$:

$$\mathbf{y} = \mathbf{C} \cdot (\overline{\mathbf{K}} \circledast \mathbf{x}) + \mathbf{D} * \mathbf{x}, \tag{XI}$$

where $\overline{\mathbf{K}}$ is generated by $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ following Eq. VI.

With the aforementioned style-aware parameters $\overline{\mathbf{A}}$ and $\mathbf{D}$ which can be expanded to global convolution kernel, we explore to generate stylized feature by a style-aware convolution. This is similar with AdaConv [4]. By doing so, the scheme adds style information to content feature efficiently while maintaining long-range dependency in content.

2. Comparison with Synchronous Mamba-based Work

We notice that there is a Mamba-based ST method: Mamba-ST [3]. The method also proposes a pipeline with mamba encoder and mamba decoder. As 2 synchronous works, our SaMam differs from Mamba-ST in following aspects.

(1) Stylized Feature Generation: Mamba-ST generate stylized features in a self-attention perspective. Following the protocol of [1, 6], in decoder, input sequance $\mathbf{X}$, weighting parameters $\mathbf{B}$ and $\mathbf{C}$ are approximately equivalent to value $\mathbf{V}$, key $\mathbf{K}$ and query $\mathbf{Q}$ of self-attention equation. Following [7, 13], Mamba-ST generate $\mathbf{B}$ and $\mathbf{C}$ from style and content features respectively. However, this self-attention solution poses a great challenge of breaking image content information. In contrast, we design our decoder from a global convolution perspective. As Mamba is demonstrated effectiveness of building long-range dependency to maintain image content details [8], we explore to introduce style information without breaking image content details by a global convolution operation on content feature. So we design a novel style-aware S6 block (S7 block). Specifically, we predict convolution kernel $\mathbf{A}$ and channel-wise scale factor $\mathbf{D}$ from style embeddings. This introduces style information to hidden state updating to acquire style selectivity. Besides, inspired by dynamical weights scheme [4, 9], The style-aware convolution kernel $\mathbf{A}$ and channel-wise scale factor $\mathbf{D}$ strike a great inference efficiency. In addition, to sufficiently utilize style information and flexibly

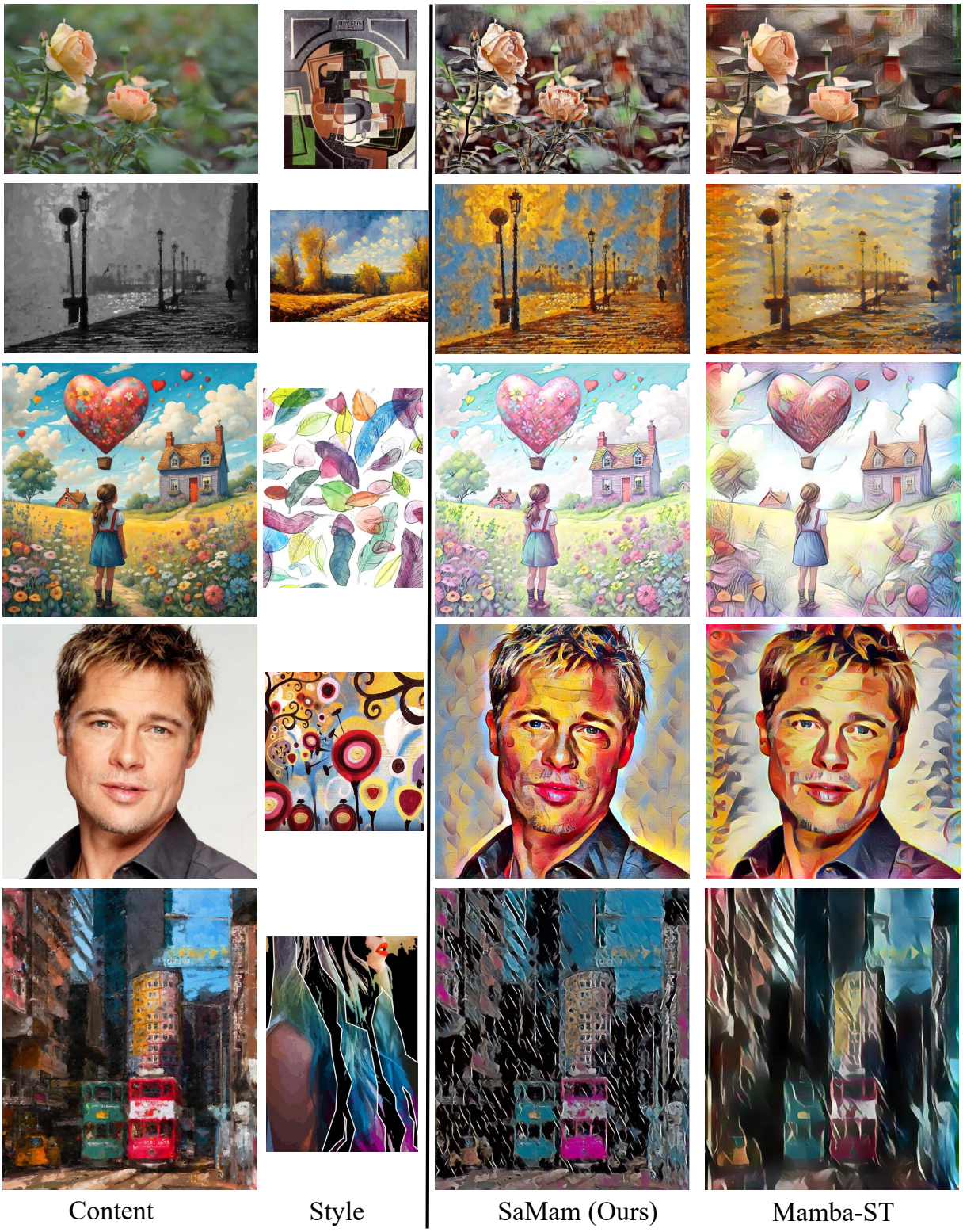

Figure I. Qualitative comparison with MambaST.

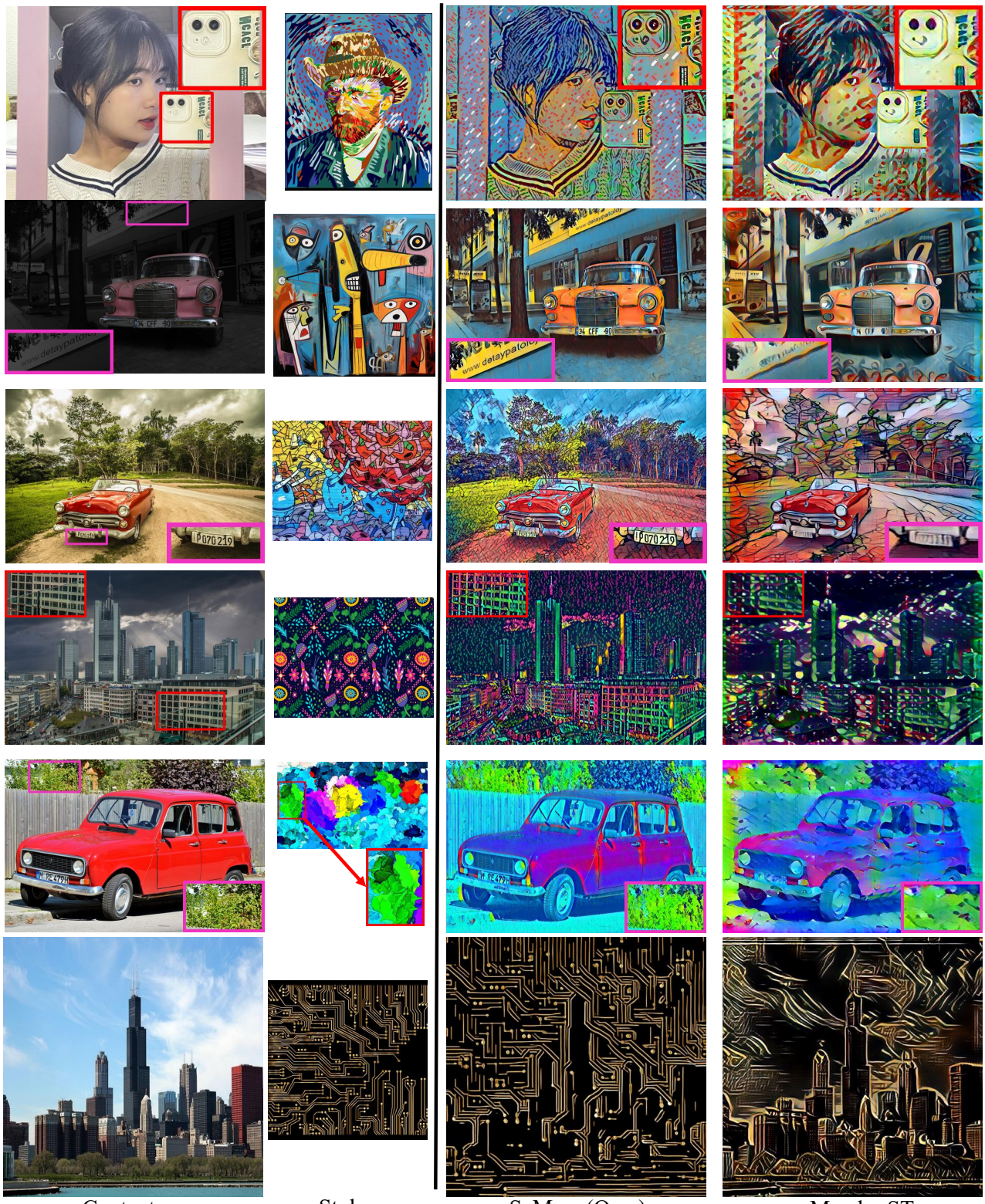

Figure II. Qualitative comparison with MambaST.

adapt to various styles, we further design SConv, SAIN and SCM in our Style-aware VSSM (SAVSSM).

(2) Zigzag Scan and Local Enhancement: Mamba-ST implements cross scan without maintaining spatial continuity. In contrast, we utilize zigzag scan to capture the continuity. Moreover, a local enhancement module is introduced to ease local pixel forgetting and channel redundancy. These 2 designs are able to promote generation quality without introducing heavy overhead during inference, thereby maintaining efficiency.

We further report the quantitative comparison of the 2 mamba-based methods. As shown in Table I, our SaMam outperforms Mamba-ST in terms of content and style quality. In addition to generation quality, our SaMam also maintains efficiency, which is highly practical for real-world applications. Finally, qualitative comparison is provided in Fig. I and Fig. II. It can be observed that Mamba-ST produces lower perceptual quality. For example, Mamba-ST is not able to capture sufficient color from style images (*e.g.*, Fig. I). Moreover, it face problems of preserving content details (*e.g.*, the 3^{rd} scene in Fig. II) and reproducing style local geometry (*e.g.*, the 5^{th} scene in Fig. II).

3. More Comparison with SOTA

3.1. Effective Receptive Field Comparison

We conduct ERF comparison with different methods. Specifically, we first randomly select 50 content images and 50 style images, then resize the size to 256×256. With the images above, we generate 2500 stylized results. Finally, we visualize averaged effective receptive field (ERF) of the center pixels. The ERF visualization corresponding to content and style images are shown in Fig. III. It can be observed that CNN based methods achieve larger ERF for content images but introduce excessively unbalanced style ERF. As Transformer based backbone is applied to capture long-range dependency, it achieves balanced ERF for style. However, Transformer based methods is not able to achieve larger ERF for content. This is because that the content features are input as query patches $\mathbf{Q}$. And there is no information interaction between the query patches in inference stage, resulting in narrow ERF for content. Moreover, Reversible-NN based method is also difficult to achieve larger ERF. In contrast, our Mamba based method achieves global ERF in terms of content and style.

3.2. More Visual Comparison

In Fig. IV, we demonstrate additional qualitative comparisons with state-of-the-art methods. It can be observed that our method is able to maintain content details and reproduce sufficient style patterns to stylized images. In contrast, other methods manipulate the content structure, miss style properties and introduce artifacts in the generated images.

Table II. User study.

Method	Content	Style	Overall
AesPA	92	75	93
S2WAT	116	67	80
CAPVST	57	88	94
Zstar	15	40	20
StyleID	53	72	57
MambaST	26	53	29
SaMam (Ours)	641	605	627

3.3. Content Leak

The content leak issue usually occurs in the stylization process because details in image content may not be sufficiently captured [2]. This type of artifact is easy to spot by human eyes after repeating several rounds of the same stylization process. We conduct experiments on this issue. The results are shown in Fig. V and Fig. VI. It can be observed that Diffusion based method StyleID [5] gradually evolves to style while losing content details excessively. As for CNN based method AesPA [10], the content structures generated after the first round are damaged (*e.g.*, the first scene in Fig. V). Although the Reversible-NN based method CAPVST [11] and Transformer based method S2WAT [12] maintain global content structures, the stylized effect is also not satisfying (*e.g.*, messy stylized image details in Fig. VI). Moreover, the mamba based methods (our SaMam and Mamba-ST [3]) are also good at preserving content structures. And our SaMam further maintain content details and produce more harmonious style textures. It's clear that SaMam effectively mitigates the content leak issue.

3.4. User Study

As a subjective task, users' preference is crucial for our method. We further conduct a user study. In the study, a single sample consists of a content image, a style image, and 7 corresponding stylization results generated by the 10 methods. We randomly select 25 content images and 25 style images to generate 25 samples for each user. For each sample, a user is asked to judge from 3 aspects: content (content fidelity), style (global color and local pattern similarity) and overall preference. Then the user votes for the one that he/she likes the most. Finally, we collect 1000 votes from 40 users. The results are shown in Table II. It can be observed that our SaMam gains best user preference.

4. Model Analysis on Network Architecture

Our model consists of three modules, including content/style mamba encoder and style-aware mamba decoder. The mamba encoder consists of successive Vision State Space Modules (VSSMs). And the decoder consists of several Style-aware Vision State Space Groups (SAVSSGs).

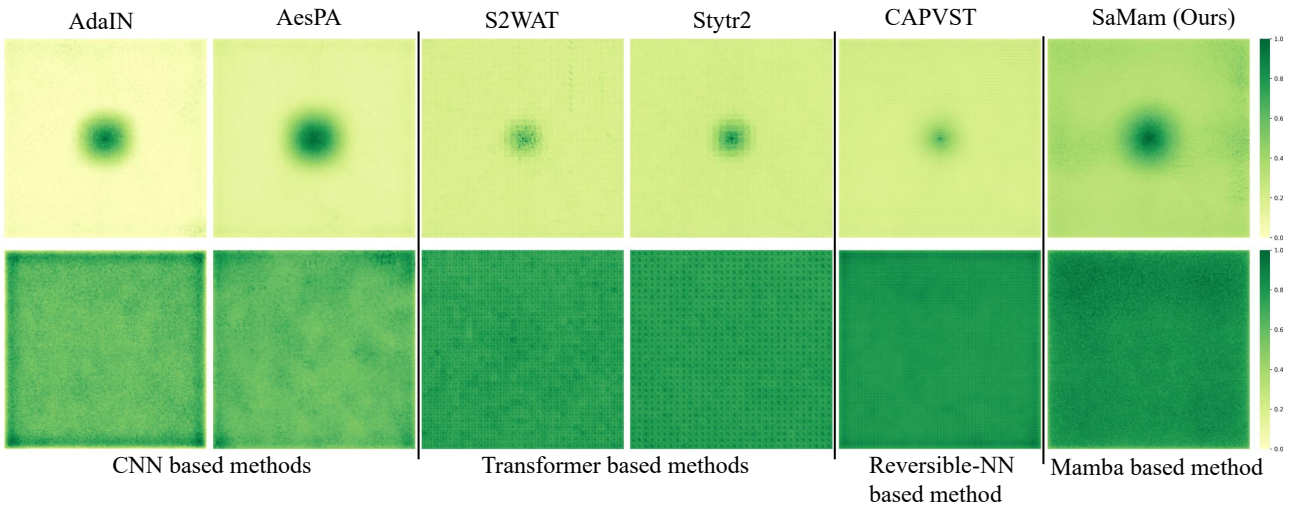

Figure III. The Effective Receptive Field (ERF) visualization. The first row is the ERF of content, while the second row shows the ERF of style.

Table III. Ablation study on model architectures.

Method	Encoder VSSMs	Decoder SAVSSMs	Decoder SAVSSGs	C	E	N	Metrics ArtFID ↓	FID ↓	LPIPS ↓	CSFD ↓
A1	4	2	2	256	512	16	26.739	18.113	0.3990	0.2990
A2	1	2	2	256	512	16	29.722	19.839	0.4263	0.3165
B1	2	4	2	256	512	16	26.411	17.986	0.3911	0.2750
B2	2	1	2	256	512	16	29.130	19.668	0.4094	0.3187
C1	2	2	4	256	512	16	**26.288**	**17.855**	0.3942	0.2735
C2	2	2	1	256	512	16	28.570	18.947	0.4323	0.2936
D1	2	2	2	512	512	16	26.302	17.914	0.3906	0.2710
D2	2	2	2	128	512	16	27.953	18.295	0.4487	0.2815
E1	2	2	2	256	768	16	26.292	17.942	**0.3880**	0.2723
E2	2	2	2	256	256	16	27.462	18.460	0.4112	0.2802
F1	2	2	2	256	512	32	26.671	18.193	0.3896	0.2838
F2	2	2	2	256	512	8	28.427	18.330	0.4706	0.2917
Ours	2	2	2	256	512	16	26.305	17.946	0.3884	**0.2703**

Specifically, each SAVSSG consists of Style-aware Vision State Space Modules (SAVSSMs). We conduct experiments to study the effect of the model capacity for these modules. Specifically, we first develop six model variants (A1-C2 in Table III) by deepening or lightening these modules. As we can see, our method does not obtain consistent performance gains as the encoders and decoder are deepened. However, our default setting produces significant gains rather than the lightweight model variants.

Moreover, we set image feature channel number C, expanded dimension size E and SSM dimension N to 256, 512 and 16 in our SaMam. We further conduct experiments to investigate the effect of the parameters (D1-F2 in Table III). It can be observed that our method achieves notable performance gains compared with D2, E2 and F2. When continuing to increase C and E, the model gains limited increase (*i.e.*, D1 and E1). Moreover, F1 even suffers from performance drop when increasing N. Consequently, $C = 256$, $E = 512$ and $N = 16$ are adopted as the default setting, which achieves the accuracy-efficiency balance.

References

[1] Ameen Ali, Itamar Zimerman, and Lior Wolf. The hidden attention of mamba models. *arXiv preprint arXiv:2403.01590*, 2024. 2

[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 Con-*

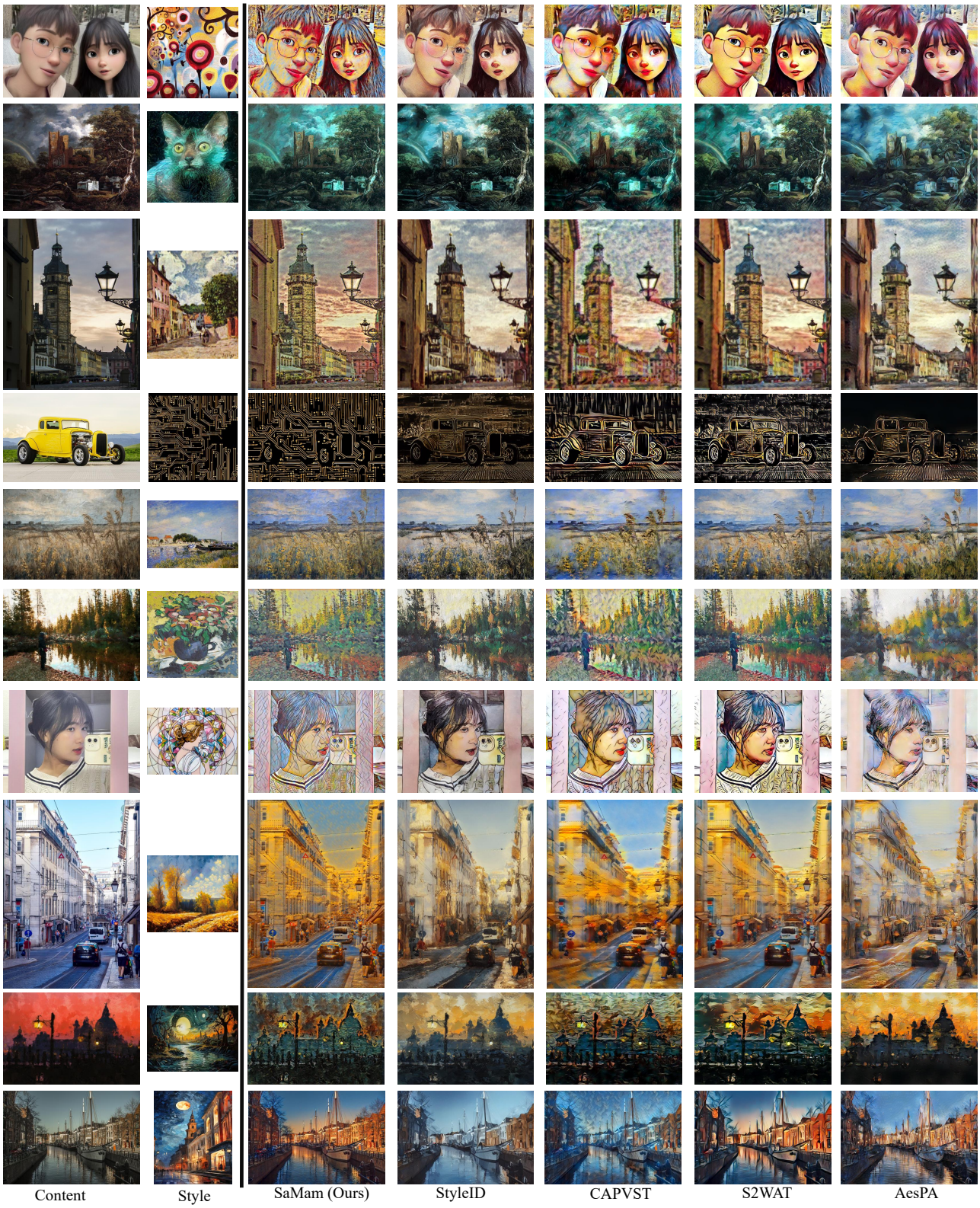

Figure IV. Qualitative comparison with SOTA.

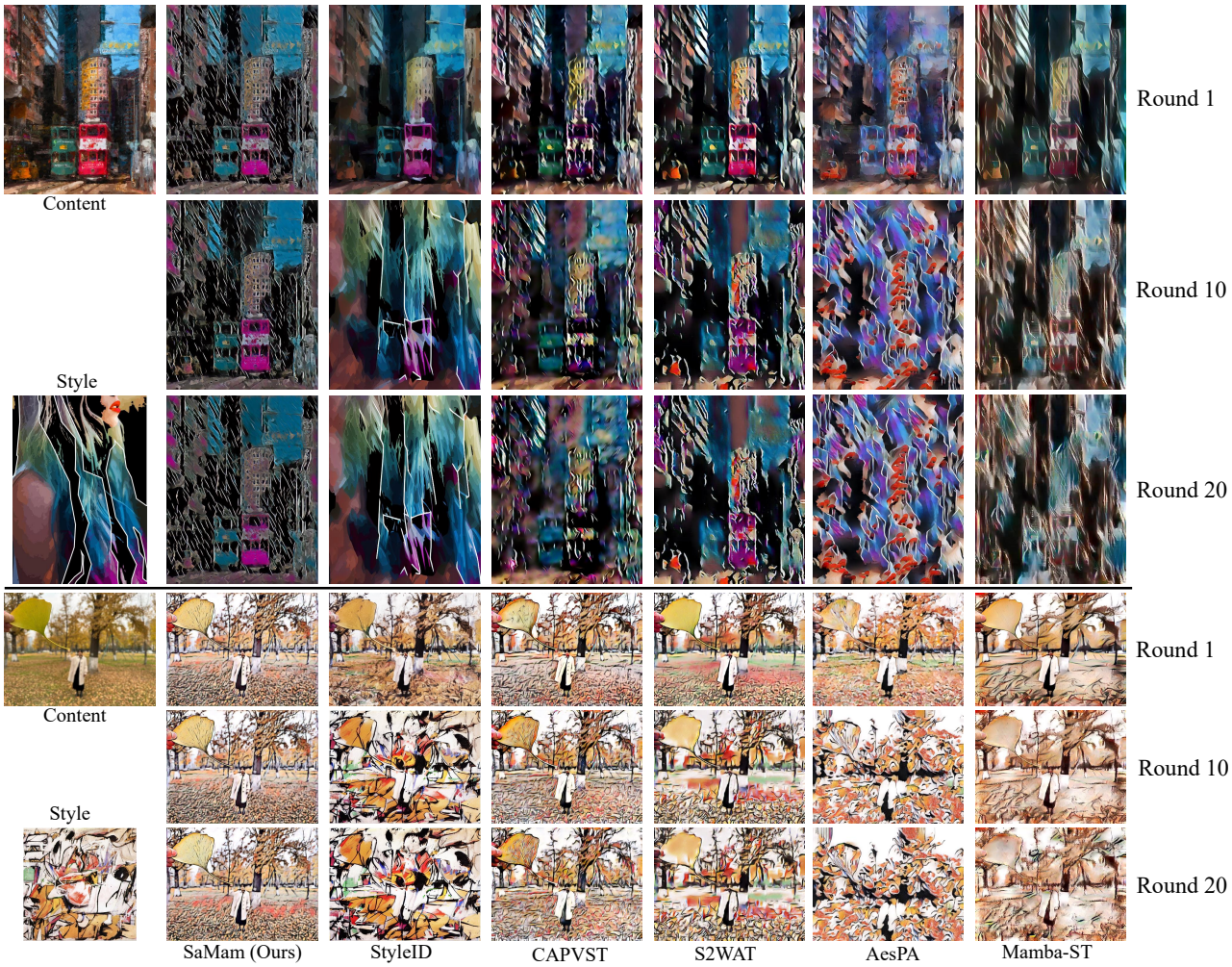

Figure V. Visual comparison of content leak issue.

ference on Computer Vision and Pattern Recognition, pages 862–871, 2021. 5

[3] Filippo Botti, Alex Ergasti, Leonardo Rossi, Tomaso Fontanini, Claudio Ferrari, Massimo Bertozzi, and Andrea Prati. Mamba-st: State space model for efficient style transfer. *arXiv preprint arXiv:2409.10385*, 2024. 2, 5

[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] 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. 5

[6] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024. 2

[7] 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. 2

[8] 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

[9] Jingwen He, Chao Dong, and Yu Qiao. 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*, pages 53–68. Springer, 2020. 2

[10] 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 Interna-*

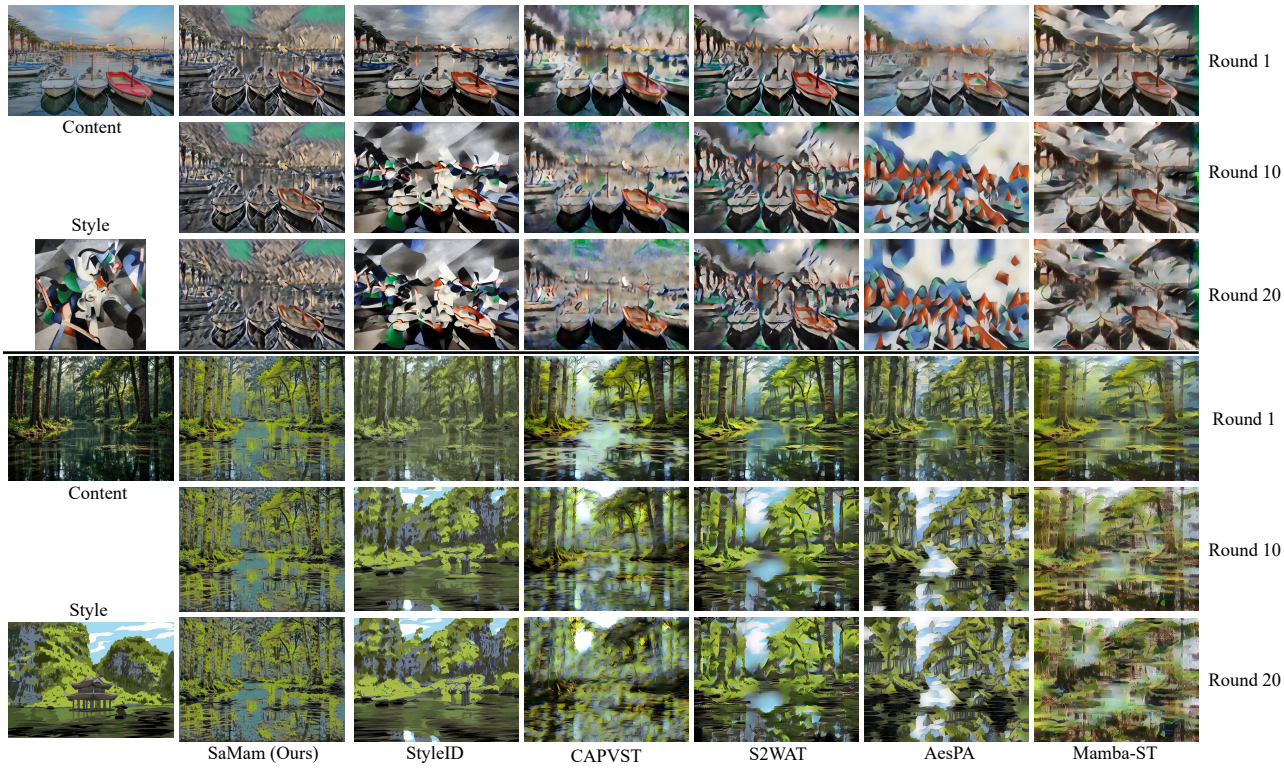

Figure VI. Visual comparison of content leak issue.

tional Conference on Computer Vision, pages 22758–22767, 2023. 5

[11] Linfeng Wen, Chengying Gao, and Changqing Zou. Cap-vstnet: content affinity preserved versatile style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18300–18309, 2023. 5

[12] Chiyu Zhang, Xiaogang Xu, Lei Wang, Zaiyan Dai, and Jun Yang. S2wat: Image style transfer via hierarchical vision transformer using strips window attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7024–7032, 2024. 5

[13] Sizhe Zheng, Pan Gao, Peng Zhou, and Jie Qin. Puff-net: Efficient style transfer with pure content and style feature fusion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8059–8068, 2024. 2