

StyTr²: Image Style Transfer with Transformers

Yingying Deng^{1,2} Fan Tang^{3*} Weiming Dong^{1,2*} Chongyang Ma⁴
Xingjia Pan² Lei Wang⁵ Changsheng Xu^{1,2}

¹ School of Artificial Intelligence, UCAS ² NLPR, Institute of Automation, CAS
³ School of Artificial Intelligence, Jilin University ⁴ Kuaishou Technology ⁵ CIPUC

Abstract

The goal of image style transfer is to render an image with artistic features guided by a style reference while maintaining the original content. Owing to the locality in convolutional neural networks (CNNs), extracting and maintaining the global information of input images is difficult. Therefore, traditional neural style transfer methods face biased content representation. To address this critical issue, we take long-range dependencies of input images into account for image style transfer by proposing a transformer-based approach called StyTr². In contrast with visual transformers for other vision tasks, StyTr² contains two different transformer encoders to generate domain-specific sequences for content and style, respectively. Following the encoders, a multi-layer transformer decoder is adopted to stylize the content sequence according to the style sequence. We also analyze the deficiency of existing positional encoding methods and propose the content-aware positional encoding (CAPE), which is scale-invariant and more suitable for image style transfer tasks. Qualitative and quantitative experiments demonstrate the effectiveness of the proposed StyTr² compared with state-of-the-art CNN-based and flow-based approaches. Code and models are available at <https://github.com/diyiyiii/StyTR-2>.

1. Introduction

Image style transfer is an interesting and practical research topic that can render a content image using a referenced style image. Based on texture synthesis, traditional style transfer methods [5, 18] can generate vivid stylized images, but are computationally complex due to the formulation of stroke appearance and painting process. Afterward, researchers focus on neural style transfer based on convolutional neural networks (CNNs). Optimization-based style transfer methods [19, 31, 47] render the input content images with learned style representation iteratively. Following the encoder-transfer-decoder pipeline, arbitrary

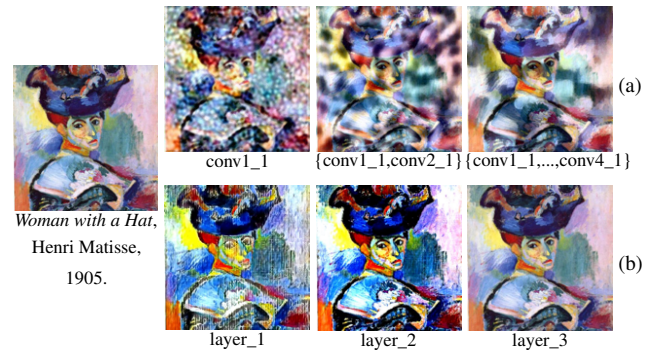


Figure 1. Comparisons of intermediate layers using the leftmost image as the input content and the style reference in a style transfer task. (a) Feature visualizations of a pretrained VGG based on Gatys *et al.* [19]. (b) Feature visualizations of our transformer decoder.

style transfer networks [2, 3, 22, 29, 30, 33, 38, 52, 54] are optimized by aligning second-order statistics of content images to style images and can generate stylized results in a feed-forward manner efficiently. However, these methods cannot achieve satisfactory results in some cases due to the limited ability to model the relationship between content and style. To overcome this issue, several recent methods [13, 14, 35, 39, 63] apply a self-attention mechanism for improved stylization results.

The aforementioned style transfer methods utilize CNNs to learn style and content representations. Owing to the limited receptive field of convolution operation, CNNs cannot capture long-range dependencies without sufficient layers. However, the increment of network depth could cause the loss of feature resolution and fine details [24]. The missing details can damage the stylization results in aspects of content structure preservation and style display. As shown in Fig. 1(a), some details are omitted in the process of convolutional feature extraction. An *et al.* [1] recently show that typical CNN-based style transfer methods are biased toward content representation by visualizing the content leak of the stylization process, i.e., after repeating several rounds of stylization operations, the extracted structures of input

*Co-corresponding authors

content will change drastically.

With the success of transformer [51] in natural language processing (NLP), transformer-based architectures have been adopted in various vision tasks. The charm of applying transformer to computer vision lies in two aspects. First, it is free to learn the global information of the input with the help of the self-attention mechanism. Thus, a holistic understanding can be easily obtained within each layer. Second, the transformer architecture models relationships in input shapes [41], and different layers extract similar structural information [46] (see Fig. 1(b)). Therefore, transformer has a strong representation capability to capture precise content representation and avoid fine detail missing. Thus, the generated structures can be well-preserved.

In this work, we aim to eliminate the biased representation issue of CNN-based style transfer methods and propose a novel image **Style Transfer Transformer** framework called **StyTr²**. Different from the original transformer, we design two transformer-based encoders in our StyTr² framework to obtain domain-specific information. Following the encoders, the transformer decoder is used to progressively generate the output sequences of image patches. Furthermore, towards the positional encoding methods that are proposed for NLP, we raise two considerations: (1) different from sentences ordered by logic, the image sequence tokens are associated with semantic information of the image content; (2) for the style transfer task, we aim to generate stylized images of any resolution. The exponential increase in image resolution will lead to a significant change of positional encoding, leading to large position deviation and inferior output quality. In general, a desired positional encoding for vision tasks should be conditioned on input content while being invariant to image scale transformation. Therefore, we propose a content-aware positional encoding scheme (**CAPE**) which learns the positional encoding based on image semantic features and dynamically expands the position to accommodate different image sizes.

In summary, our main contributions include:

- A transformer-based style transfer framework called StyTr², to generate stylization results with well-preserved structures and details of the input content image.
- A content-aware positional encoding scheme that is scale-invariant and suitable for style transfer tasks.
- Comprehensive experiments showing that StyTr² outperforms baseline methods and achieves outstanding results with desirable content structures and style patterns.

2. Related Work

Image style transfer. Gatys *et al.* [19] find that hierarchical layers in CNNs can be used to extract image content structures and style texture information and propose an

optimization-based method to generate stylized images iteratively. Some approaches [25, 28] adopt an end-to-end model to achieve real-time style transfer for one specific style. For more efficient applications, [7, 17, 32] combine multiple styles in one model and achieve outstanding stylization results. More generally, arbitrary style transfer gains more attention in recent years. Huang *et al.* [22] propose an adaptive instance normalization (AdaIN) to replace the mean and variance of content with that of style. AdaIN is widely adopted in image generation tasks [2, 21, 26, 33, 52] to fuse the content and style features. Li *et al.* [29] design a whiten and colorization transformation (WCT) to align the second-order statistics of content and style features. Moreover, many methods [1, 50, 57] also aim at promoting the generation effect in the premise of efficiency. Based on the CNNs model, [13, 14, 35, 39, 56] introduce self-attention to the encoder-transfer-decoder framework for better feature fusion. Chen *et al.* [9] propose an Internal-External Style Transfer algorithm (IEST) containing two types of contrastive loss, which can produce a harmonious and satisfactory stylization effect. However, existing encoder-transfer-decoder style transfer methods cannot handle the long-range dependencies and may lead to missing details.

Transformer for vision tasks. As an alternative to recurrent and convolutional neural networks, transformer [51] is first proposed for machine translation tasks and has been widely used in various NLP tasks [4, 11, 15, 36, 43, 44]. Inspired by the breakthrough of transformer in NLP, many researchers have developed vision transformers for various image/video related tasks [60], including object detection [6, 12, 65], semantic segmentation [53, 64], image classification [10, 16, 37, 55, 61], image processing and generation [8, 10, 24]. Compared with fully convolutional networks, transformer-based networks can capture long-term dependencies of the input image by using self-attention mechanisms. In this paper, we introduce transformer-based structures for style transfer tasks which can be seen as sequence-to-sequence generation of image patches.

Positional encoding. Positional encoding is commonly used in transformer-based models to provide position information. There are two types of positional encoding are used: *functional* and *parametric* positional encoding. Functional positional encoding is calculated by pre-defined functions, such as sinusoidal functions [51]. Parametric positional encoding is learned via model training [15]. To ensure translational-invariance for the transformers, relative positional encoding [20, 45, 48, 62] considers the distance between tokens in the image sequence. [59] and [23] further include positional encoding in CNN-based models as spatial inductive. In this paper, we propose a content-aware positional encoding mechanism that is scale-invariant and more suitable for image generation tasks.

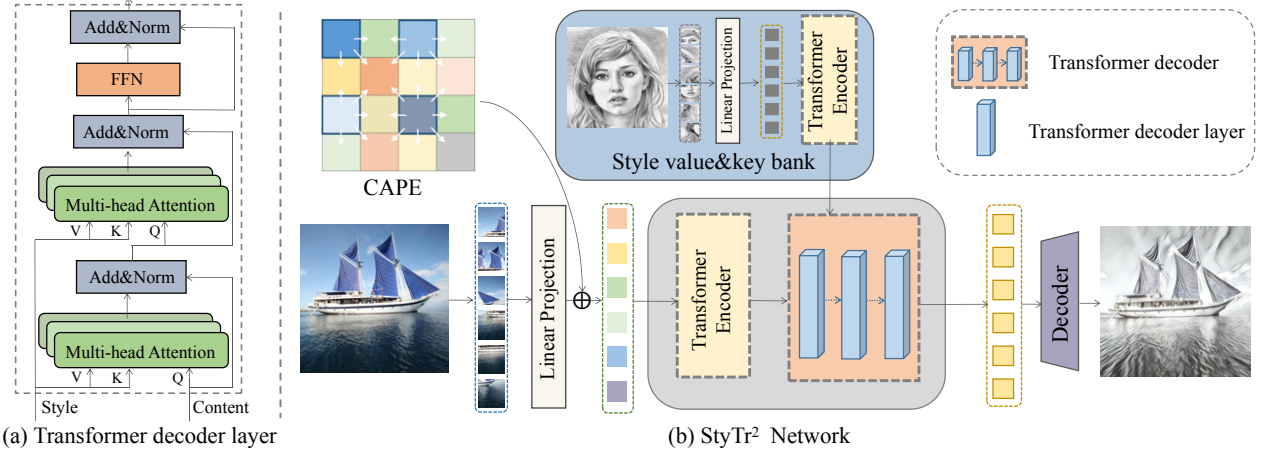


Figure 2. Overall pipeline of our StyTr² framework. We split the content and style images into patches and use a linear projection to obtain patch sequences. Then, the content sequences added with CAPE are fed into a content transformer encoder, while the style sequences are fed into a style transformer encoder. Following the two transformer encoders, a multi-layer transformer decoder is adopted to stylize the content sequences according to the style sequences. Finally, we use a progressive upsampling decoder to obtain the final output.

3. Our Method

To leverage the capability of transformers to capture long-range dependencies of image features for style transfer, we formulate the problem as a sequential patch generation task. Given a content image $I_c \in \mathbb{R}^{H \times W \times 3}$ and a style image $I_s \in \mathbb{R}^{H \times W \times 3}$, we split both images into patches (similar to tokens in NLP tasks) and use a linear projection layer to project input patches into a sequential feature embedding \mathcal{E} in a shape of $L \times C$, where $L = \frac{H \times W}{m \times m}$ is the length of \mathcal{E} , $m = 8$ is the patch size and C is the dimension of \mathcal{E} . The overall structure of our framework is shown in Fig. 2.

3.1. Content-Aware Positional Encoding

When using a transformer-based model, the positional encoding (PE) should be included in the input sequence to acquire structural information. According to [51], the attention score of the i -th patch and the j -th patch is computed as:

$$\begin{aligned}
 A_{i,j} &= ((\mathcal{E}_i + \mathcal{P}_i)W_q)^T ((\mathcal{E}_j + \mathcal{P}_j)W_k) \\
 &= W_q^T \mathcal{E}_i^T \mathcal{E}_j W_k + W_q^T \mathcal{E}_i^T \mathcal{P}_j W_k \\
 &\quad + W_q^T \mathcal{P}_i^T \mathcal{E}_j W_k + W_q^T \mathcal{P}_i^T \mathcal{P}_j W_k,
 \end{aligned} \tag{1}$$

where W_q and W_k are parameter matrices for query and key calculation, and \mathcal{P}_i presents the i -th one-dimensional PE. In 2D cases, the positional relative relation between the patch at a pixel (x_i, y_i) and the patch at a pixel (x_j, y_j) is:

$$\begin{aligned}
 &\mathcal{P}(x_i, y_i)^T \mathcal{P}(x_j, y_j) \\
 &= \sum_{k=0}^{\frac{d}{4}-1} [\cos(w_k(x_j - x_i)) + \cos(w_k(y_j - y_i))],
 \end{aligned} \tag{2}$$

where $w_k = 1/10000^{2k/128}$, $d = 512$. The positional relative relation between two patches only depends on

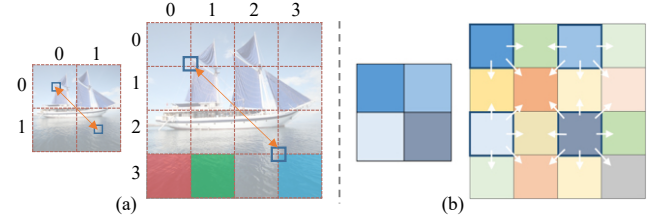


Figure 3. Schematic diagram of content-aware positional encoding.

their spatial distance. Accordingly, we raise two important questions. *First, for an image generation task, should we take image semantics into account when calculating PE?* Traditional PE is designed for sentences ordered by logic, but image patches are organized based on the content. We denote the distance between two patches as $d(\cdot, \cdot)$. On the right-hand side of Fig. 3(a), the difference between $d((x_0, y_3), (x_1, y_3))$ (the red and green patches) and $d((x_0, y_3), (x_3, y_3))$ (the red and cyan patches) should be small because we expect similar content patches to have similar stylization results. *Second, is the traditional sinusoidal positional encoding still suitable for vision tasks when the input image size expands exponentially?* As shown in Fig. 3(a), when an image is resized, the relative distance between patches (depicted by small blue rectangles) in the same locations can change dramatically, which may be not suitable for multi-scale methods in vision tasks.

To this end, we propose content-aware positional encoding (CAPE), which is scale-invariant and more suitable for style transfer tasks. Different from sinusoidal PE which only considers the relative distance of patches, CAPE is conditioned on the semantics of image content. We assume that using $n \times n$ positional encodings is adequate to represent the semantics of an image. For an image $I \in \mathbb{R}^{H \times W \times 3}$, we rescale the fixed $n \times n$ positional encoding to $\frac{H}{m} \times \frac{W}{m}$, as

shown in Fig. 3(b). In this way, various image scales will not influence the spatial relation between two patches. The CAPE of patch (x, y) , namely, $\mathcal{P}_{CA}(x, y)$, is formulated as

$$\begin{aligned} \mathcal{P}_{\mathcal{L}} &= \mathcal{F}_{\text{pos}}(\text{AvgPool}_{n \times n}(\mathcal{E})), \\ \mathcal{P}_{CA}(x, y) &= \sum_{k=0}^s \sum_{l=0}^s (a_{kl} \mathcal{P}_{\mathcal{L}}(x_k, y_l)), \end{aligned} \quad (3)$$

where $\text{AvgPool}_{n \times n}$ is the average pooling function, \mathcal{F}_{pos} is 1×1 convolution operation used as a learnable positional encoding function, $\mathcal{P}_{\mathcal{L}}$ is learnable PE following the sequence \mathcal{E} , n is set to 18 in our experiments, a_{kl} is the interpolation weight, and s is the number of neighboring patches. Lastly, we add \mathcal{P}_{CA_i} to \mathcal{E}_i as the final feature embedding of the i -th patch at a pixel location (x, y) .

3.2. Style Transfer Transformer

Transformer encoder. We capture long-range dependencies of image patches by using transformer based structure to learn sequential visual representations. Different from other vision tasks [6, 8, 12], the input of the style transfer task comes from two different domains, corresponding to natural images and artistic paintings, respectively. Therefore, StyTr^2 has two transformer encoders to encode domain-specific features, which are used to translate a sequence from one domain to another in the next stage.

Given the embedding of an input content sequence $Z_c = \{\mathcal{E}_{c1} + \mathcal{P}_{CA1}, \mathcal{E}_{c2} + \mathcal{P}_{CA2}, \dots, \mathcal{E}_{cL} + \mathcal{P}_{CAL}\}$, we first feed it into the transformer encoder. Each layer of the encoder consists of a multi-head self-attention module (MSA) and a feed-forward network (FFN). The input sequence is encoded into query (Q), key (K), and value (V):

$$Q = Z_c W_q, \quad K = Z_c W_k, \quad V = Z_c W_v, \quad (4)$$

where $W_q, W_k, W_v \in \mathbb{R}^{C \times d_{head}}$. The multi-head attention is then calculated by

$$\begin{aligned} \mathcal{F}_{\text{MSA}}(Q, K, V) &= \text{Concat}(\text{Attention}_1(Q, K, V), \\ &\dots, \text{Attention}_N(Q, K, V))W_o, \end{aligned} \quad (5)$$

where $W_o \in \mathbb{R}^{C \times C}$ are learnable parameters, N is the number of attention heads, and $d_{head} = \frac{C}{N}$. The residual connections are applied to obtain the encoded content sequence Y_c :

$$\begin{aligned} Y'_c &= \mathcal{F}_{\text{MSA}}(Q, K, V) + Q, \\ Y_c &= \mathcal{F}_{\text{FFN}}(Y'_c) + Y'_c, \end{aligned} \quad (6)$$

where $\mathcal{F}_{\text{FFN}}(Y'_c) = \max(0, Y'_c W_1 + b_1) W_2 + b_2$. Layer normalization (LN) is applied after each block [51].

Similarly, the embedding of an input style sequence $Z_s = \{\mathcal{E}_{s1}, \mathcal{E}_{s2}, \dots, \mathcal{E}_{sL}\}$ is encoded into a sequence Y_s following the same calculation process, except that positional encoding is not considered because we do not need to maintain structures of the input style in the final output.

Transformer decoder. Our transformer decoder is used to translate the encoded content sequence Y_c according to the encoded style sequence Y_s in a regressive fashion. Different from the auto-regressive process in NLP tasks, we take all the sequential patches as input at one time to predict the output. As shown in Fig. 3(a), each transformer decoder layer contains two MSA layers and one FFN. The input of our transformer decoder includes the encoded content sequence, i.e., $\hat{Y}_c = \{Y_{c1} + \mathcal{P}_{CA1}, Y_{c2} + \mathcal{P}_{CA2}, \dots, Y_{cL} + \mathcal{P}_{CAL}\}$, and the style sequence $Y_s = \{Y_{s1}, Y_{s2}, \dots, Y_{sL}\}$. We use the content sequence to generate the query Q , and use the style sequence to generate the key K and the value V :

$$Q = \hat{Y}_c W_q, \quad K = Y_s W_k, \quad V = Y_s W_v. \quad (7)$$

Then, the output sequence X of the transformer decoder can be calculated by

$$\begin{aligned} X'' &= \mathcal{F}_{\text{MSA}}(Q, K, V) + Q, \\ X' &= \mathcal{F}_{\text{MSA}}(X'' + \mathcal{P}_{CA}, K, V) + X'', \\ X &= \mathcal{F}_{\text{FFN}}(X') + X'. \end{aligned} \quad (8)$$

Layer normalization (LN) is also applied at the end of each block [51].

CNN decoder. The output sequence X of the transformer is in a shape of $\frac{HW}{64} \times C$. Instead of directly upsampling the output sequence to construct the final results, we use a three-layer CNN decoder to refine the outputs of the transformer decoder following [64]. For each layer, we expand the scale by adopting a series of operations including 3×3 Conv + ReLU + $2 \times$ Upsample. Finally, we can obtain the final results in a resolution of $H \times W \times 3$.

3.3. Network Optimization

The generated results should maintain the original content structures and the reference style patterns. Therefore, we construct two different perceptual loss terms to measure the content difference between the output image I_o and the input content image I_c , as well as the style difference between I_o and the input style reference I_s .

We use feature maps extracted by a pretrained VGG model to construct the content loss and the style loss following [1, 22]. The content perceptual loss \mathcal{L}_c is defined as

$$\mathcal{L}_c = \frac{1}{N_l} \sum_{i=0}^{N_l} \|\phi_i(I_o) - \phi_i(I_c)\|_2, \quad (9)$$

where $\phi_i(\cdot)$ denotes features extracted from the i -th layer in a pretrained VGG19 and N_l is the number of layers.

Resolution	Ours	StyleFormer	IEST	AdaAttN	ArtFlow	MCC	MAST	AAMS	SANet	Avatar	AdaIN
256×256	0.116	0.013	0.065	0.104	0.142	0.013	0.030	2.074	0.015	0.260	0.007
512×512	0.661	0.026	0.092	0.213	0.418	0.015	0.096	2.173	0.019	0.470	0.008

Table 1. Average inference time (in seconds) of different methods at two output resolutions.

The style perceptual loss \mathcal{L}_s is defined as

$$\mathcal{L}_s = \frac{1}{N_l} \sum_{i=0}^{N_l} \|\mu(\phi_i(I_o)) - \mu(\phi_i(I_s))\|_2 + \|\sigma(\phi_i(I_o)) - \sigma(\phi_i(I_s))\|_2, \quad (10)$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and variance of extracted features, respectively.

We also adopt identity loss [39] to learn richer and more accurate content and style representations. Specifically, we take two of the same content (style) images into StyTr², and the generated output $I_{cc}(I_{ss})$ should be identical to the input $I_c(I_s)$. Therefore, we compute two identity loss terms to measure the differences between $I_c(I_s)$ and $I_{cc}(I_{ss})$:

$$\begin{aligned} \mathcal{L}_{id1} &= \|I_{cc} - I_c\|_2 + \|I_{ss} - I_s\|_2, \\ \mathcal{L}_{id2} &= \frac{1}{N_l} \sum_{i=0}^{N_l} \|\phi_i(I_{cc}) - \phi_i(I_c)\|_2 + \|\phi_i(I_{ss}) - \phi_i(I_s)\|_2. \end{aligned} \quad (11)$$

The entire network is optimized by minimizing the following function:

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s + \lambda_{id1} \mathcal{L}_{id1} + \lambda_{id2} \mathcal{L}_{id2}. \quad (12)$$

We set λ_c , λ_s , λ_{id1} , and λ_{id2} to 10, 7, 50, and 1 to alleviate the impact of magnitude differences.

4. Experiments

4.1. Implementation Details

MS-COCO [34] is used as the content dataset and WikiArt [42] is used as the style dataset. In the training stage, all the images are randomly cropped into a fixed resolution of 256×256 , while any image resolution is supported at the test time. We adopt the Adam optimizer [27] and the learning rate is set to 0.0005 using the warm-up adjustment strategy [58]. We set the batch size to be 8 and train our network with 160,000 iterations.

4.2. Comparisons with SOTA Methods

We compare our method with AdaIN [22], Avatar [49], SANet [39], AAMS [63], MAST [14], MCC [13], ArtFlow [1], AdaAttN [35], IEST [9], and StyleFormer [56]. AdaIN, Avatar, SANet, AAMS, and MAST are typical CNN-based image stylization approaches. MCC [13] is a video

style transfer method but can be applied to images without damaging the generated results. ArtFlow [1] designs a flow-based network to minimize image reconstruction error and recovery bias. AdaAttN [35] performs attentive normalization on a per-point basis for feature distribution alignment. IEST [9] takes advantage of contrastive learning and external memory to boost visual quality. StyleFormer [56] adopts the transformer mechanism into the traditional CNN-based encoder-decoder pipeline. By contrast, we present a *pure* transformer-based architecture to solve the issue of missing content details caused by convolutions.

Timing information. Our model is trained on two NVIDIA Tesla P100 GPUs and two NVIDIA GeForce RTX 3090 GPUs for approximately one day. In Table 1, we compare the inference time of different style transfer methods under two output resolutions using one Tesla P100.

Qualitative evaluation. Fig. 4 shows the visual results of qualitative comparisons. Owing to the simplified alignment of mean and variance, the results of AdaIN [22] have insufficient style patterns. The stylized images present crack artifacts that affect the overall transfer quality. AAMS [63] focuses on the main structure (referring to salient regions in the attention map) of the content image but ignores the other parts. Therefore, the secondary structures are not well maintained. The patch-swap-based method leads to artifacts of over-blurry output. MCC [13] uses a transform formulation of self-attention, but the absence of non-linear operation limiting the maximum value of network output results in an overflow issue around object boundaries. The flow-based model has limited capability of feature representation, thus the results of ArtFlow [1] generally have the problem of insufficient or inaccurate style. The border of stylized images may present undesirable patterns due to numerical overflow. The per-point basis of AdaAttN [35] leads to style degeneration, thus the stylized patterns in the generated results are not consistent with the input reference. The visual quality of IEST [9] outperforms other approaches. However, the style of generated results may not be consistent with the input style reference (the 1st and 3rd rows). Following the CNN-based ‘‘encoder-decoder’’ pipeline, results of StyleFormer [56] still tend to missing details. By contrast, StyTr² leverages a transformer-based network, which has better feature representation to capture long-range dependencies of input image features and to avoid



Figure 4. Qualitative comparisons of style transfer results using different methods.

	Ours	StyleFormer	IEST	AdaAttN	ArtFlow	MCC	MAST	AAMS	SANet	Avatar	AdaIN
$\mathcal{L}_c \downarrow$	1.91	2.86	<u>1.97</u>	2.29	2.13	2.38	2.46	2.44	2.44	2.84	2.34
$\mathcal{L}_s \downarrow$	<u>1.47</u>	2.91	3.47	2.45	3.08	1.56	1.55	3.18	1.18	2.86	1.91

Table 2. Quantitative comparisons. We compute the average content and style loss values of results by different methods to measure how well the input content and style are preserved. The best results are in **bold** while the second-best results are marked with an underline.

missing of content and style details. Therefore, our results can achieve well-preserved content structures and desirable style patterns.

Quantitative evaluation. We calculate the *content difference* between the generated results and input content images as well as the *style difference* between the generated results and input style images, as two indirect metrics of the style transfer quality. Intuitively, the smaller the difference the better the input content/style is preserved. We randomly select 40 style images and 20 content images to generate 800 stylized images. For each method, we compute the content difference based on Eq. (9) and calculate the style difference following Eq. (10). Table 2 shows the corresponding quantitative results. Overall, our method achieves the lowest content losses and IEST [9] is the second-best. However, as discussed in the qualitative evaluation above, the style loss of IEST is the highest because the style appearance of generated results is not far from the input style reference. In terms of style loss, SANet [39] and StyTr² outperform the other

methods. Therefore, our results can effectively preserve both the input content and the reference style simultaneously.

4.3. Analysis of Content Leak

The content leak issue usually occurs in the stylization process because CNN-based feature representation may not sufficiently capture details in the image content. This type of artifact is easy to spot by human eyes after repeating several rounds of the same stylization process [1], which is formulated by

$$I_o^i = G_i(\dots G_2(G_1(I_c, I_s), I_s) \dots, I_s), \quad (13)$$

where G_i is the generator for the i -th round and I_o^i is the corresponding stylization result. To solve the content leak problem, An *et al.* [1] propose a reversible network to replace CNN-based models. However, strict reversibility may not be suitable for generation tasks [40]. Furthermore, the robustness and generated visual effects of ArtFlow may be downgraded due to limited capability of feature

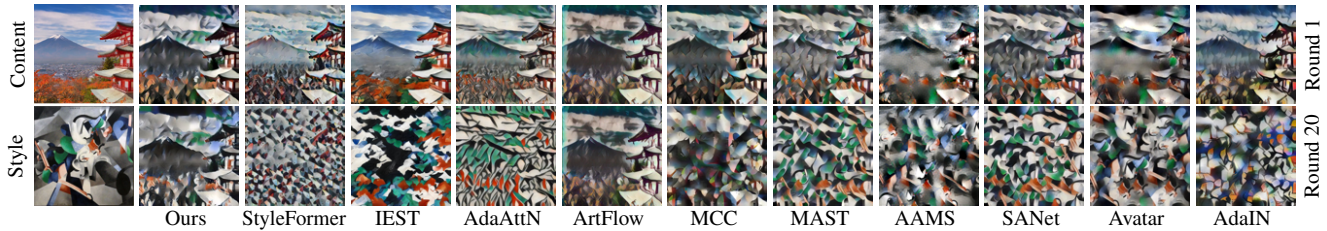


Figure 5. Visualization of the content leak issue. Top/bottom: results after running the same stylization process using a certain method after the 1st and the 20th round, respectively.

representation. By contrast, we leverage the capability of transformer-based architecture to capture long-range dependencies. Thus, our method can significantly alleviate the content leak issue.

We compare StyTr² with CNN-based methods and flow-based model ArtFlow [1]. Fig. 5 demonstrates the corresponding results after the 1st and the 20th rounds of the repeating stylization process. As shown in the top row, the content structures generated by CNN-based methods after the first round are damaged to various degrees, but our result still presents clear content details. Although the results generated by ArtFlow maintain clear content structures, the stylized effect is not satisfactory (e.g., the marginal flaws and inappropriate style patterns). The bottom row of Fig. 5 shows that with increasing rounds of the stylization process, the content structures generated by CNN-based methods tend to be blurry, while the content structures generated by our method remain distinct. The same problem applies to StyleFormer, which also relies on the CNN-based encoder-decoder pipeline. Therefore, our model captures precise content representation leading to superior style transfer results while effectively alleviating the content leak issue.

4.4. Analysis of CAPE

As described in Section 3.1, when calculating PE, we should take the semantic information of content images into account. To compare the proposed CAPE with sinusoidal PE which is not semantics-aware, we show two cases where the input content image has repetitive patterns or is simply collaged by repeating one image four times. As shown in Fig. 6, we can observe inconsistent stylized regions in the final results when using sinusoidal PE. The input resolution is set to be 256×256 , which is the same as the image resolution for training.

Moreover, handling input resolution different from the training examples is generally challenging for a learning-based method. To this end, an ideal PE for vision tasks should be scale-invariant, but a drastic change of image resolution leads to a significant difference in traditional PE. We compare our CAPE with sinusoidal PE in Fig. 7. In the third row, the input size is 512×512 , which is twice the image resolution for training. Consequently, the results present vertical track artifacts due to the large



Figure 6. Comparisons of sinusoidal PE and CAPE using content images with repetitive patterns.

positional deviation. In the second row, the input resolution is 256×256 , which is the same as the training data. The corresponding results do not have the issue of vertical tracks but are not satisfactory due to the small resolution. By contrast, our method supports any input resolution with CAPE by design. Therefore, our results in the last row of Fig. 7 present clear content structures and proper stylized patterns. Additional ablation studies are provided in our supplementary materials.

To verify whether CAPE could provide position information, we show CAPEs with different inputs in Fig. 8. Although two CAPEs are not the same, they have similar encoding behaviors such as highlighted diagonal, repeat, and periodic patterns as learnable PE [16] and sinusoidal PE. Different from learnable PE where the encoding is conditioned on the whole dataset, our CAPE dynamically encodes different input and thus can easily generalize to various resolutions.

4.5. User study

We conduct a user study to further compare our method. AdaAttN [35], ArtFlow [1], MCC [13], AAMS [63] and AdaIN [22] are selected as baselines. We reuse the images in the quantitative comparison and invite 100 participants to evaluate the results of different approaches. The participants are informed of the purpose and details of our user study. The participants comprise 52 males and 48 females, in the age range of 19~45. Given a content image and a style image, we show the result generated by our approach and the output from another randomly selected method for comparison and



Figure 7. Comparisons of sinusoidal PE and CAPE using content images with two different resolutions.

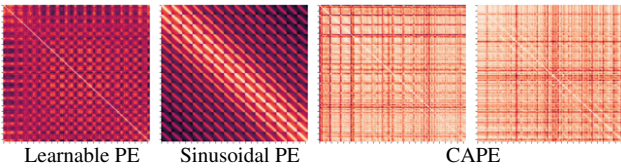


Figure 8. Normalized attention scores of different types of PE.

ask the user to choose (1) which result has better stylization effects (2) which stylization result preserves the content structures better and (3) which stylization result transfers the style patterns more consistently. No time limit is given for the selection process, but the assessment time should be longer than 30 seconds for each question. Each participant conducts 40 rounds of comparisons and we collect 4,000 votes for each question. We count the votes that existing methods are preferred to ours and show the statistical results in Table 3. Our method is superior to other approaches in all three criteria of overall quality, content preservation, and style consistency.

5. Conclusion

In this work, we propose a novel framework called StyTr², for image style transfer. Our StyTr² includes a content transformer encoder and a style transformer encoder to capture domain-specific long-range information. A transformer decoder is developed to translate the content

	AdaAttN	ArtFlow	MCC	AAMS	AdaIN
Overall	44.8%	25.7%	25.5%	18.2%	23.5%
Content	45.7%	28.3%	20.6%	19.4%	31.7%
Style	23.6%	16.7%	27.8%	14.8%	13.5%

Table 3. User study results. Each number represents the percentage of votes that the corresponding method is preferred to ours, using the criteria of overall quality, preservation of content and style, respectively.

sequences based on the reference style sequences. We also propose a content-aware positional encoding scheme that is semantics-aware and is suitable for scale-invariant visual generation tasks. As the first baseline for style transfer using a visual transformer, StyTr² alleviates the content leak problem of CNN-based models and provides fresh insight into the challenging style transfer problem. At present, the test-time speed of our method is not as fast as some CNN-based approaches. Incorporating some priors from CNNs to speed up the computation would be an interesting future direction.

Acknowledgment. This work was supported by National Key R&D Program of China under no. 2020AAA0106200, by National Natural Science Foundation of China under nos. 61832016, U20B2070, 6210070958, 62102162, by CASIA-Tencent Youtu joint research project, and by Open Projects Program of National Laboratory of Pattern Recognition.

References

- [1] Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. ArtFlow: Unbiased image style transfer via reversible neural flows. In *IEEE/CVF Conferences on Computer Vision and Pattern Recognition (CVPR)*, pages 862–871, 2021. [1](#), [2](#), [4](#), [5](#), [6](#), [7](#)
- [2] Jie An, Tao Li, Haozhi Huang, Li Shen, Xuan Wang, Yongyi Tang, Jinwen Ma, Wei Liu, and Jiebo Luo. Real-time universal style transfer on high-resolution images via zero-channel pruning. *arXiv preprint arXiv:2006.09029*, 2020. [1](#), [2](#)
- [3] Jie An, Haoyi Xiong, Jun Huan, and Jiebo Luo. Ultrafast photorealistic style transfer via neural architecture search. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 10443–10450, 2020. [1](#)
- [4] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1877–1901, 2020. [2](#)
- [5] Stefan Bruckner and M Eduard Gröller. Style transfer functions for illustrative volume rendering. *Computer Graphics Forum*, 26(3):715–724, 2007. [1](#)
- [6] 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 (ECCV)*, pages 213–229, 2020. [2](#), [4](#)
- [7] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1897–1906, 2017. [2](#)
- [8] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [2](#), [4](#)
- [9] Haibo Chen, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, Dongming Lu, et al. Artistic style transfer with internal-external learning and contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. [2](#), [5](#), [6](#)
- [10] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning (ICML)*, pages 1691–1703, 2020. [2](#)
- [11] Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019. [2](#)
- [12] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [2](#), [4](#)
- [13] Yingying Deng, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, and Changsheng Xu. Arbitrary video style transfer via multi-channel correlation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 1210–1217, 2021. [1](#), [2](#), [5](#), [7](#)
- [14] Yingying Deng, Fan Tang, Weiming Dong, Wen Sun, Feiyue Huang, and Changsheng Xu. Arbitrary style transfer via multi-adaptation network. In *ACM International Conference on Multimedia*, pages 2719–2727, 2020. [1](#), [2](#), [5](#)
- [15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. [2](#)
- [16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. [2](#), [7](#)
- [17] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *International Conference on Learning Representations (ICLR)*, 2016. [2](#)
- [18] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of Annual Conference on Computer Graphics and Interactive Techniques*, pages 341–346, 2001. [1](#)
- [19] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. [1](#), [2](#)
- [20] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations (ICLR)*, 2021. [2](#)
- [21] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *European Conference on Computer Vision (ECCV)*, pages 172–189, 2018. [2](#)
- [22] Xun Huang and Belongie Serge. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1501–1510, 2017. [1](#), [2](#), [4](#), [5](#), [7](#)
- [23] Md Amirul Islam, Sen Jia, and Neil D. B. Bruce. How much position information do convolutional neural networks encode? In *International Conference on Learning Representations (ICLR)*, 2020. [2](#)
- [24] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. TransGAN: Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. [1](#), [2](#)
- [25] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016. [2](#)
- [26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019. [2](#)

- [27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. [5](#)
- [28] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision (ECCV)*, pages 702–716, 2016. [2](#)
- [29] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances Neural Information Processing Systems (NeurIPS)*, pages 386–396, 2017. [1](#), [2](#)
- [30] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *European Conference on Computer Vision (ECCV)*, pages 453–468, 2018. [1](#)
- [31] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017. [1](#)
- [32] Minxuan Lin, Fan Tang, Weiming Dong, Xiao Li, Changsheng Xu, and Chongyang Ma. Distribution aligned multimodal and multi-domain image stylization. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 17(3):96:1–96:17, 2021. [2](#)
- [33] Tianwei Lin, Zhuoqi Ma, Fu Li, Dongliang He, Xin Li, Errui Ding, Nannan Wang, Jie Li, and Xinbo Gao. Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [1](#), [2](#)
- [34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. [5](#)
- [35] 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 International Conference on Computer Vision (ICCV)*, pages 6649–6658, 2021. [1](#), [2](#), [5](#), [7](#)
- [36] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. [2](#)
- [37] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [2](#)
- [38] Ming Lu, Hao Zhao, Anbang Yao, Yurong Chen, Feng Xu, and Li Zhang. A closed-form solution to universal style transfer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5952–5961, 2019. [1](#)
- [39] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5880–5888, 2019. [1](#), [2](#), [5](#), [6](#)
- [40] Taesung Park, Alexei A. Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, 2020. [6](#)
- [41] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. [2](#)
- [42] Fred Phillips and Brandy Mackintosh. Wiki art gallery, inc.: A case for critical thinking. *Issues in Accounting Education*, 26(3):593–608, 2011. [5](#)
- [43] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *Preprint*, 2018. [2](#)
- [44] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. [2](#)
- [45] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. [2](#)
- [46] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Advances Neural Information Processing Systems (NeurIPS)*, 2021. [2](#)
- [47] Eric Risser, Pierre Wilmot, and Connelly Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint arXiv:1701.08893*, 2017. [1](#)
- [48] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018. [2](#)
- [49] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *Computer Vision and Pattern Recognition (CVPR)*, 2018 *IEEE Conference on*, pages 1–9, 2018. [5](#)
- [50] Jan Svoboda, Asha Anoopsh, Christian Osendorfer, and Jonathan Masci. Two-stage peer-regularized feature recombination for arbitrary image style transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13816–13825, 2020. [2](#)
- [51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. [2](#), [3](#), [4](#)
- [52] Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, and Ming-Hsuan Yang. Collaborative distillation for ultra-resolution universal style transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1860–1869, 2020. [1](#), [2](#)
- [53] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [2](#)
- [54] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7789–7798, 2020. [1](#)
- [55] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020. [2](#)
- [56] Xiaolei Wu, Zhihao Hu, Lu Sheng, and Dong Xu. Style-former: Real-time arbitrary style transfer via parametric style composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14618–14627, 2021. [2](#), [5](#)
- [57] Zhijie Wu, Chunjin Song, Yang Zhou, Minglun Gong, and Hui Huang. Efanet: Exchangeable feature alignment network for arbitrary style transfer. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 12305–12312, 2020. [2](#)
- [58] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tiejian Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning (ICML)*, pages 10524–10533, 2020. [5](#)
- [59] Rui Xu, Xintao Wang, Kai Chen, Bolei Zhou, and Chen Change Loy. Positional encoding as spatial inductive bias in gans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [2](#)
- [60] Yifan Xu, Huapeng Wei, Minxuan Lin, Yingying Deng, Kekai Sheng, Mengdan Zhang, Fan Tang, Weiming Dong, Feiyue Huang, and Changsheng Xu. Transformers in computational visual media: A survey. *Computational Visual Media*, 8(1):33–62, 2022. [2](#)
- [61] Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-ViT: Slow-fast token evolution for dynamic vision transformer. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. [2](#)
- [62] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. [2](#)
- [63] Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, and Jun Wang. Attention-aware multi-stroke style transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1467–1475, 2019. [1](#), [5](#), [7](#)
- [64] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [2](#), [4](#)
- [65] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021. [2](#)