# *Two Birds, One Stone*: A Unified Framework for Joint Learning of Image and Video Style Transfers

Bohai Gu[1,3] Heng Fan[2] Libo Zhang[1,3†]
[1] Institute of Software Chinese Academy of Sciences, Beijing, China
[2] Department of Computer Science and Engineering, University of North Texas, Denton TX, USA
[3] University of Chinese Academy of Sciences, Beijing, China

## Abstract

*Current arbitrary style transfer models are limited to either image or video domains. In order to achieve satisfying image and video style transfers, two different models are inevitably required with separate training processes on image and video domains, respectively. In this paper, we show that this can be precluded by introducing **UniST**, a **Uni**fied **S**tyle **T**ransfer framework for both images and videos. At the core of UniST is a domain interaction transformer ($DIT$), which first explores context information within the specific domain and then interacts contextualized domain information for joint learning. In particular, $DIT$ enables exploration of temporal information from videos for the image style transfer task and meanwhile allows rich appearance texture from images for video style transfer, thus leading to mutual benefits. Considering heavy computation of traditional multi-head self-attention, we present a simple yet effective axial multi-head self-attention (AMSA) for $DIT$, which improves computational efficiency while maintains style transfer performance. To verify the effectiveness of UniST, we conduct extensive experiments on both image and video style transfer tasks and show that UniST performs favorably against state-of-the-art approaches on both tasks. Code is available at `https://github.com/NevSNev/UniST`.*

## 1. Introduction

Artistic image style transfer [13] aims at migrating a desirable style pattern from an inference image to the origin image while preserving the original content structures. Although CNNs based methods have been well studied in this field [16, 21, 22, 29], they fail to capture the long-range interaction between the style and content domains, which may result in suboptimal results.

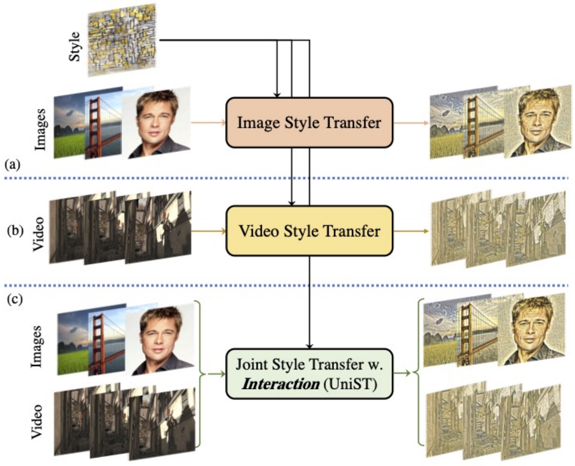


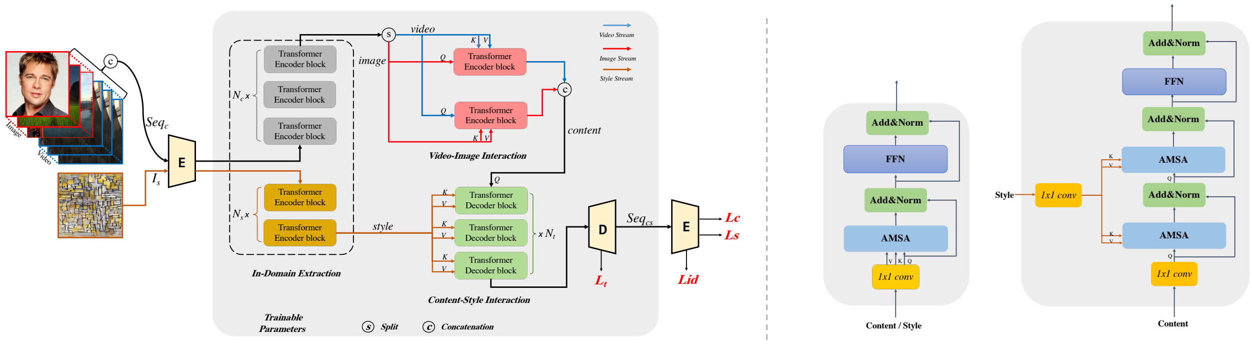
Figure 1: Comparison between single domain (image or video) style transfer and our joint style transfer.

Recently, owing to the ability to model long-range dependencies, Transformers [31] have shown excellent performance in a wide range of tasks including style transfer. For example, Stytr$^2$ [9] introduces a pure transformer network to deal with image style transfer. However, pixel-level self-attention brings additional computational complexity, resulting in lower efficiency.

Unlike image style transfer, video style transfer brings in new challenges of preserving temporal consistency between stylized video frames. To achieve style transfer on the video domain, a feasible solution is to adapt existing image-based style transfer methods (*e.g.* [21, 25]) by retraining them with modification in architecture and/or loss functions. Despite simplicity, this domain adaption requires another repetitive and tedious training process, resulting in resource waste to some extent. Some other methods (*e.g.* [7, 38]) directly adopt the same model from video to image, but the results are somewhat visually flawed.

To solve the above issues, we present a **Uni**fied **S**tyle

[†] Corresponding author: Libo Zhang (libo@iscas.ac.cn).



(a) Unified Style Transfer Framework

(b) Transformer Encoder Block

(c) Transformer Decoder Block

Figure 2: (a) Overview of the UniST, where the $E$ is the VGG-19 network (pretrained and fixed) and $D$ is the CNN decoder with a symmetric structure of VGG-19. $\mathcal{L}_c$, $\mathcal{L}_s$, $\mathcal{L}_{id}$ and $\mathcal{L}_t$ are content loss, style loss, identity loss and temporal loss; (b) The structure of improved transformer encoder block; (c) The structure of improved transformer decoder block.

**T**ransfer framework, termed **UniST**, for both images and videos. The proposed network leverages the local and long-range dependencies jointly. More specifically, UniST first applies CNNs to generate tokens, and then models long-range dependencies to excavate domain-specific information with domain interaction transformer ($DIT$). Afterwards, $DIT$ sequentially interacts contextualized domain information for joint learning. Considering that the vanilla self-attention suffers from a heavy computational burden, we are inspired by axial attention [32] and develop the Axial Multi-head Self-Attention (AMSA) mechanism to calculate attention efficiently for either images or video input. To our best knowledge, our approach is the first unified solution to handle both image and video style transfers simultaneously.

To verify the effectiveness of our approaches, we carry out extensive experiments on ImageNet [6] and MPI [2] for image and video field respectively. The results demonstrate that our unified solution can achieve better performance than current state-of-the-art image-based and video-based algorithms, evidencing its superiority and efficiency.

In summary, we make the following contributions in this work: (1) We propose a new joint learning framework for arbitrary image and video style transfers, in which two tasks can benefit from each other to improve the performance. To our best knowledge, this is the first work towards a unified solution with joint interaction. (2) We develop the Axial Multi-head Self-Attention mechanism to address computational complexity and adapt to tokens from image and video input. (3) Extensive experiments on both image and video style transfer tasks demonstrate the effectiveness of our approach compared with state-of-the-art methods.

## 2. Related Work

**Image Style Transfer** CNNs based style transfer models are widely applied in the image field. Gatys *et al*. [13] apply the CNN model to iteratively generate stylized outputs. Johnson *et al*. [18] adopt an end-to-end model to accomplish real-time style transfer for the specific style. More generally, fast arbitrary style transfer is attracting enormous attention. Therefore, Huang *et al*. [16] achieve arbitrary style transfer by adaptively applying mean and standard deviation of style to that of content (AdaIN), which is widely adopted in image generation tasks [19, 23] for better feature fusion. Similarly, Li *et al*. [22] accomplish style transfer with two transformation steps including whitening and coloring. Then, Sheng *et al*. [29] design a multi-scale model combined with AdaIN and style-swap.

Recently, [4, 7, 8, 25–27, 37] introduce the self-attention mechanism to the encoder-transfer-decoder framework for better style transfer. Moreover, Deng *et al*. [9] take advantage of the Transformer's long-range dependencies while refusing to adopt the CNN's local dependencies. Meanwhile, the pure transformer is hugely computational and the position encoding needs to be presented specially. All of the above lead to slow inference speed. In contrast, UniST leverages both the Transformer's long-range dependencies and CNN's locality dependencies to build a unified framework for joint learning of image and video style transfers. After one pass of training, our framework can generate vivid image and video stylization results in real-time applications.

**Video Style Transfer** In addition to image style transfer, video style transfer presents new challenges, including both vivid stylization results and well-maintained temporal consistency. To this end, many previous works [3, 12, 15, 39] directly add optical flows consistency constraint to image style transfer solutions to enhance the inter-frame correlation. However, the optical flow requires extra complex computation, making it impractical to process high-resolution or long videos. So there emerge some works that addresses the stability issue with other approaches instead of optical flow warping. Li *et al*. [21] present a linear transformation module which based on content and style features. And Wu *et al*. [36] propose a SANet based framework that addresses the temporal consistency with a SSIM consistency constraint. Moreover, Deng *et al*. [7] learn the per-channel correlation via a transformed self-attention. On this routine, Liu *et al*. [25] improve the temporal consistency via a modified self-attention and a cross-image similarity loss. Similarly, Wu *et al*. [38] devise a generic contrastive coherence preserving loss applied to local patches. Despite meeting multi-task domains, the results are somewhat flawed.

In this work, without using any inter-frame information like optical flows, UniST takes advantage of the unified image-video joint learning style transfer framework to facilitate video stylization effects. And the experiments demonstrate that it also achieves great temporal consistency.

## 3. Methodology

**Overall Framework.** Given a style image $I_s \in R^{H \times W \times 3}$ and the content sequence $Seq_c \in R^{T \times H \times W \times 3}$, which is the concatenation of image and video. Our framework eventually synthesizes the stylized sequence $Seq_{cs} \in R^{T \times H \times W \times 3}$. As in Figure 2, our framework leverages local and long-range dependencies jointly. Notably, the CNN encoder is not only used for local spatial information extraction, but also for tokenization. Then we use $DIT$ to accomplish two types of style transfer task jointly, which first explores context information within the content and style domains and then interacts contextualized domain information for joint learning. Below, we will detail our framework.

### 3.1. Tokenization

As discussed above, Deng *et al*. [9] split the input images into patches directly for transformer input with the necessary position encoding, resulting in low efficiency. In this work, we take advantage of CNNs to strengthen locality and improve the efficiency. Similar to [16], we use the pre-trained VGG-19 network [30] to extract feature maps of input images. Then, the 512-dim vector at each pixel in the $relu4\_1$ layer is treated as a token for further transformer encoders. In this way, we combine CNNs and transformer to exploit both local and long-range dependencies. Meanwhile, there is no need to maintain the position encoding.

### 3.2. Domain Interaction Transformer

As in Figure 2, $DIT$ consists of three modules in turn, namely intra-domain extraction, video-image and style-content interaction. Specifically, the domain-specific extraction module is stacked with $N_c$, $N_s$ transformer encoder blocks for the $Seq_c$ and $I_s$ respectively. While content-style interaction is stacked with $N_t$ transformer decoder blocks.

**In-Domain Extraction** Based on the tokenization, the local context has already been extracted. So $DIT$ further exploits domain-specific information with a number of consecutively stacked transformer encoder blocks in Figure 2(b). Taking either $Seq_c$ or $I_s$ as input, the in-domain extraction module simultaneously capture the long-range information within the content and style domains. Normally, the transformer encoder block consists of a multi-head self-attention (MSA) layer and a feed-forward network (FFN). For the better efficiency, we develop the Axial Multi-head Self-Attention (AMSA) mechanism to replace MSA, combined with a $1 \times 1$ convolutional layer for locality strengthening. And this mechanism is applied to all of the transformer blocks mentioned below. In addition, residual connections and layer normalization are deployed after each layer. The transformer encoder block is defined as:

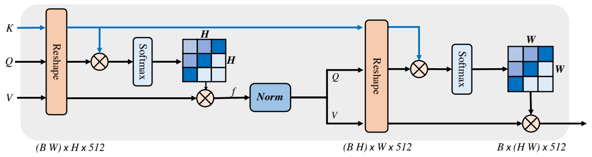


Figure 3: The structure of the AMSA. $Norm$ here denotes the layer normalization.

$$\mathcal{S}' = \mathcal{AMSA}(\text{Conv}(Q), \text{Conv}(K), \text{Conv}(V)) + Q, \quad (1)$$

$$\mathcal{S} = \mathcal{FFN}(\mathcal{S}') + \mathcal{S}', \quad (2)$$

where $\mathcal{S}$ is the output sequence.

**Video-Image Interaction.** After domain-specific excavation, $DIT$ presents a symmetric module based on two transformer encoder blocks to interact contextual information between two types of content modalities. Notably, we split the input content sequence into two parts: one half is video $Seq_v$ and the other half is image $Seq_I$. As illustrated in Figure 2, one of the blocks takes the $Seq_v$ as $Q$, the $Seq_I$ as $K$, $V$, and the other one does the opposite. In this way, joint learning is performed for style transfer. And two types of content sequence are concatenated after interaction.

**Content-Style Interaction.** To finally capture the relevance between the content and the style domain, this module consists of a set of stacked transformer decoder blocks. As in Figure 2(c), each transformer decoder block takes the content as $Q$ while the style as the $K$ and $V$, alternately containing two ASMA layers and one FFN layer. Similarly, layer normalization and residual connections are applied after each layer. The transformer decoder block is defined as:

$$\mathcal{S}^{''} = \mathcal{AMSA}(\text{Conv}(Q), \text{Conv}(K), \text{Conv}(V)) + Q, \quad (3)$$

$$\mathcal{S}' = \mathcal{AMSA}(\text{Conv}(\mathcal{S}^{''}), \text{Conv}(K), \text{Conv}(V)) + \mathcal{S}^{''}, \quad (4)$$

$$\mathcal{S} = \mathcal{FFN}(\mathcal{S}') + \mathcal{S}'. \quad (5)$$

### 3.3. Axial Multi-head Self-Attention.

Inspired by [32], $DIT$ computes self-attention along a separate axis, rather than in feature maps like other transformer models [9,11,14,35,40]. The improved AMSA layer is specialized for style transfer without position encoding.

An AMSA layer consists of two MSA [31] layers in total, operating sequentially along the height and width axes. As in Figure 3, we first reshape $Q$, $K$, $V$ to separate the width dimension information, then use the first MSA layer to calculate self-attention along the height axis, and the output $f$ is normalized using the layer normalization ($LN$) layer. Following almost the same process as before, except that what we separate in the second MSA layer is the height dimension. Notably, we use the previous output $f$ as $Q$ and $V$, while keeping $K$ unchanged. Experiments in Table 4 verify this dedicated AMSA layer reduces the memory consumption and meanwhile improves inference efficiency.

The reasons behind our AMSA are two-fold. First, the information contained in the previous $K, V$ is crucial for the second MSA layer to learn contextual dependencies either within or across domains. Besides, the $V$ of the second MSA contains the last axial information, which is critical to ensure the stability of stylization. Experiments in Figure 10 show that our design effectively prevents artifacts caused by side effects of axial attention in style transfer.

### 3.4. Unimodal input for inference.

In inference, UniST takes bimodal content input and outputs both style transfer results simultaneously, achieving *two birds with one stone*. Since the model has learned complementary content knowledge, UniST could also take unimodal content input into account by replacing the cross-attention with self-attention in the video-image interaction without compromising the quality of results. As in Table 6, unimodality obtains the consistent scores with bimodality.

### 3.5. Loss functions.

The overall loss function is the weighted summation of the content loss $\mathcal{L}_c$, style loss $\mathcal{L}_s$, identity loss $\mathcal{L}_{\text{id}}$ and temporal loss $\mathcal{L}_t$:

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s + \mathcal{L}_{\text{id}} + \lambda_t \mathcal{L}_t, \tag{6}$$

where $\lambda_c$, $\lambda_s$ and $\lambda_t$ are balancing factors. UniST uses the pre-trained VGG-19 to extract feature maps as $\phi = \{relu1\_1, relu2\_1, relu3\_1, relu4\_1\}$. In our experiment, we use the above four layers with equal weights to calculate the loss below.

We use Euclidean distance [10] to compute content loss $\mathcal{L}_c$:

$$\mathcal{L}_c = \sum_{i=1}^{4} \left\| \phi_i(Seq_{\text{cs}}) - \phi_i(Seq_c) \right\|_2 . \tag{7}$$

Following [16], the style loss $\mathcal{L}_s$ is defined as:

$$\begin{aligned}\mathcal{L}_s = \sum_{i=1}^{4} \big( &\|\mu(\phi_i(Seq_{\text{cs}})) - \mu(\phi_i(I_s))\|_2 \\ &+ \|\sigma(\phi_i(Seq_{\text{cs}})) - \sigma(\phi_i(I_s))\|_2 \big),\end{aligned} \tag{8}$$

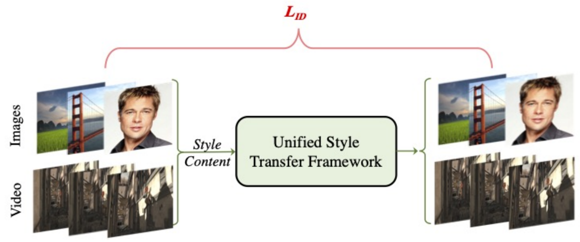


Figure 4: Illustration of the identity loss.

where $\mu(\cdot)$ and $\sigma(\cdot)$ denote the mean and variance of features separately. We further use the identity loss $\mathcal{L}_{id}$ [27] to promote more accurate content and style representation:

$$\begin{aligned}\mathcal{L}_{\text{id}} &= \lambda_{\text{id1}}(\|Seq_{\text{cc}} - Seq_c\|_2 + \|I_{\text{ss}} - I_s\|_2) \\ &+ \lambda_{\text{id2}}(\sum_{i=1}^{4} \|\phi_i(Seq_{\text{cc}}) - \phi_i(Seq_c)\|_2 + \|\phi_i(I_{\text{ss}}) - \phi_i(I_s)\|_2),\end{aligned} \tag{9}$$

where $\lambda_{\text{id1}}$ and $\lambda_{\text{id2}}$ both are balancing factors. $Seq_{\text{cc}}/I_{\text{ss}}$ denote the results synthesized from two identical content sequences or style images. Figure 4 illustrates the identity loss for better understanding.

Following the recent work AdaAttN [25], we preserve the temporal consistency via a cross-image temporal loss. This loss promotes the cosine distance $D_{cs}$ between adjacent stylization frames to be closer to the cosine distance $D_c$ between adjacent origin frames.

$$\mathcal{L}_t = \sum_{i=3}^{4} \phi_i\Big(\frac{1}{N_{c_1} N_{c_2}} \sum_{m,n} \Big| \frac{D^{m,n}_{c_1,c_2}}{\sum_m D^{m,n}_{c_1,c_2}} - \frac{D^{m,n}_{cs_1,cs_2}}{\sum_m D^{m,n}_{cs_1,cs_2}} \Big| \Big), \tag{10}$$

where $D^{m,n}_{u,v} = 1 - \frac{F^m_u \cdot F^n_v}{\|F^m_u\| \times \|F^n_v\|}$. $N$ is the spatial dimension of the current feature map. $D^{m,n,x}_{u,v}$ measures cosine distance, and $F^k$ represents the feature vector of the $k$-th entry. Note, we only adopt layer $relu3\_1$ and $relu4\_1$ to calculate $\mathcal{L}_t$. Meanwhile, in each training iteration, we compute the temporal loss between the consecutive video frames. Noted that temporal information is implicitly encoded in this way.

## 4. Experiment

### 4.1. Implementing Details

UniST is trained with MS-COCO [24] (image), MPI [2] (video) as the content datasets and WikiArt [28] as the style datasets. For $Seq_c$, the ratio of the two types of content is $1:1$, and the total number is 6. In the training phase, all images are loaded with the resolution of $256 \times 256$. While in the inference phase, UniST can be applied to any resolutions. Therefore, we manage to extend the UniST to multi-granularity style transfer (see supplementary material). In experiments, we adopt the resolutions of $1024 \times 1024$ and $1024 \times 2048$ for image and video respectively. For the loss function, the balancing factors $\lambda_c$, $\lambda_s$, $\lambda_t$, $\lambda_{\text{id1}}$, $\lambda_{\text{id2}}$ are set as 0.1, 1.5, 90, 0.1, 0.5, empirically. The number of transformer blocks $N_c$, $N_s$, $N_t$ are empirically set as $2, 1, 3$. Our

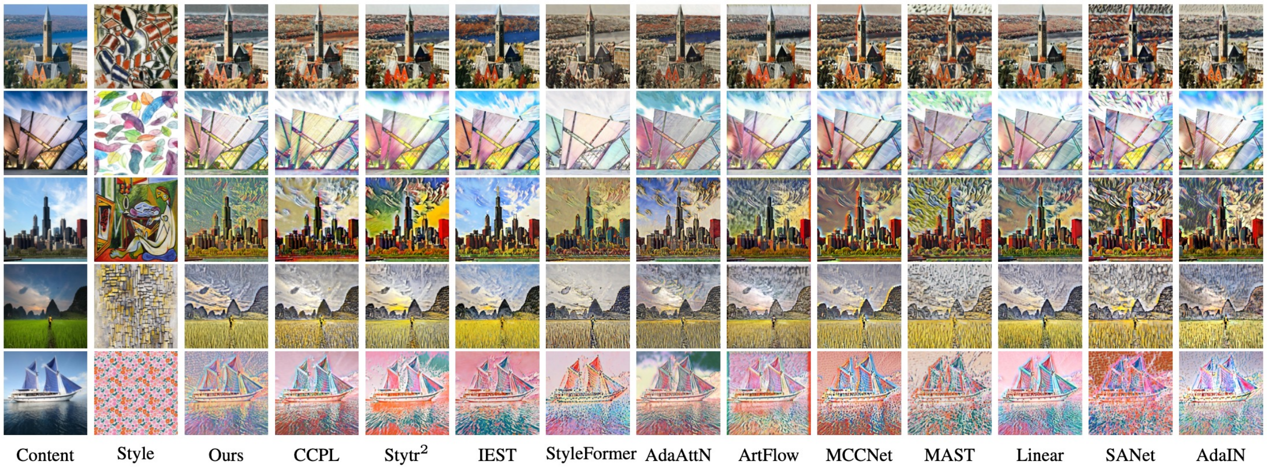


Figure 5: Qualitative comparison in image style transfer. Please zoom in for better view. Additional vivid stylization results are provided in our supplementary materials.

network is trained for $50K$ iterations on a NVIDIA Tesla V100 GPU with a batch size of 3. We use the Ranger optimizer [34] with initial learning rate of $0.00005$.

### 4.2. Image Style Transfer

We compare our method with 13 state-of-the-art methods, including AdaIN [16], SANet [27], Linear [21], MCCNet [7], MAST [8], Artflow [1], AdaAttN [25], StyleFormer [37], IEST [4], Stytr$^2$ [9], CCPL [38], epsAM [5], MicroAST [33], MCCNetV2 [20].

**Quantitative comparison.** We randomly collect $17,124$ content images from ImageNet [6] and $17,124$ style images from WikiArt [28] which are separated from the training set to generate $17,124$ stylization results. Similar to Stytr$^2$ [9], we use the mean euclidean distance and the mean instance statistics difference mentioned in section 3.5 as metrics for content preservation and stylization degree. Furthermore, we conduct the color distribution experiments by adopting color loss in DSLR [17] and adopt the Gram matrices [13] for texture difference. As in Table 1, compared with existing methods, UniST achieves the best performance in both content and style differences and obtains promising results on texture and color differences of style aspects.

**Qualitative comparison.** As in Figure 5, AdaIN [16] transfers the style patterns but losses important content details (1st, 2nd rows). SANet [27] fails to align the distribution of the style patterns, leading to the distorted object boundaries (1st, 3rd, 5th rows) and inconsistent content background structures (2nd, 4th rows). The stylization of Linear [21] is not satisfactory enough, resulting relatively light migration effects such as pink in the 5th row. As with the linear transformation, MCCNet [7] learns the correlation of each channel through the transformed self-attention, slightly improving the transfer effect, but there are serious overflow problems around the object boundaries (1st, 3rd, 4th rows). MAST [8] distorts the content background structure with excessive style transfer (3rd, 4th rows). Based on WCT [22] in our setting, Artflow [1] leads to conspicuous vertical artifacts at the edges of the generated results(3rd, 5th rows). AdaAttN [25] does the decent foreground style transfer, but requires further rendering of the background (3rd, 5th rows). StyleFormer [37] provides vivid style patterns, but the colors of the results are inconsistent with the style reference images (1st, 3rd, 4th rows). On the contrary, IEST [4] presents stylization results with consistent colors, lacking more desirable style patterns (3rd, 4th, 5th rows). Stytr$^2$ [9] fails to preserve the background structure of the content inference image (2nd, 5th rows). CCPL [38] transfers the style patterns with a lot of vertical artifacts (1st, 3rd, 4th rows). In contrast, based on the inductive bias of CNNs, our method captures the relevance between content and style sequences jointly and efficiently. As a result, we provide vivid stylized results with desirable style pattern details, while keeping the content structure well-maintained.

### 4.3. Video Style Transfer

For video style transfer, we compare our method with four state-of-the-art methods including Linear [21], MCCNet [7], AdaAttN [25], and CCPL [38]. Note, optical flow is not used for stabilization when conducting comparison.

**Quantitative comparison.** Following Liu *et al.* [25], we adopt the official optical flow [2] to wrap the output stylized frame and compute the per-pixel difference between warped and stylized frames. Meanwhile, we use the LPIPS [41] to measure the diversity of adjacent stylized frames, with smaller values indicating better consistency.

In practise, the style input is fixed at $512 \times 512$, and the content input is fixed at $256 \times 256$. Table 2 presents optical

| Methods | Ours | epsAM | MicroAST | MCCNetv2 | CCPL | StyleFormer | IEST | Stytr$^2$ | AdaAttN | MCCNet | Artflow | MAST | SANet | Linear | AdaIN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{D}_C(\downarrow)$ | **12.36** | 20.30 | 13.18 | 16.15 | 14.66 | 16.82 | 15.38 | 13.67 | 14.54 | 15.51 | <u>12.55</u> | 17.83 | 17.96 | 15.04 | 18.42 |
| $\mathcal{D}_S(\downarrow)$ | **0.46** | 1.03 | 0.92 | 1.06 | 0.71 | 0.80 | 1.38 | 0.50 | 1.07 | 0.74 | 0.85 | 0.60 | <u>0.47</u> | 0.61 | 0.52 |
| $\mathcal{TD}(\downarrow)$ | <u>69.69</u> | 155.54 | 177.20 | 143.68 | 163.00 | 115.84 | 241.69 | 89.93 | 173.87 | 107.91 | 169.39 | 94.58 | 74.39 | 87.72 | **66.95** |
| $\mathcal{CD}(\downarrow)$ | <u>17894</u> | **16546** | 21128 | 20292 | 21971 | 18344 | 23953 | 21543 | 21739 | 20566 | 19618 | 20241 | 19811 | 20948 | 21120 |

Table 1: Quantitative comparison in image style transfer. The best two results are highlighted in bold and underline.

flow error and LPIPS metrics for 20 styles over 23 videos of compared methods. UniST achieves the best scores in both metrics and thus has the best temporal consistency.

| Methods | Optical flow error(↓) | | | | LPIPS(↓) | | | | $\mathcal{D}_S(\downarrow)$ |
|---|---|---|---|---|---|---|---|---|---|
| | Style1 | Style2 | Style3 | Mean | Style1 | Style2 | Style3 | Mean | |
| Ours | **3.64** | **6.16** | **5.78** | **3.86** | **1.73** | **2.05** | **2.04** | **1.79** | **13.70** |
| AdaAttN (*ICCV 2021*) | 4.26 | 7.09 | 6.71 | 3.91 | 2.26 | 2.49 | 2.46 | 2.05 | 14.56 |
| MCCNet (*AAAI 2021*) | 4.60 | 6.83 | 6.50 | 4.57 | 2.13 | 2.36 | 2.34 | 2.07 | 15.21 |
| Linear (*CVPR 2019*) | 4.23 | 6.81 | 7.10 | 4.25 | 2.08 | 2.27 | 2.26 | 2.02 | 14.70 |
| CCPL (*ECCV 2022*) | 5.14 | 7.65 | 7.57 | 4.90 | 2.10 | 2.33 | 2.30 | 2.06 | 14.69 |

Table 2: The optical flow error ($\times 10^{-2}$) and LPIPS of SOTAs using 20 styles. Smaller values mean better temporal consistency. For clarity, only three styles are presented. Please refer to supplementary material for more details.

**Qualitative comparison.** Figure 7 shows the results of the video qualitative comparison. To better verify long-term temporal consistency, we take the 5-th, 15-th and 25-th frames of the example video as the reference content images, where the character in the video has large movements. The results show that all four methods visually satisfy the long-term temporal consistency. To be specific, Linear [21] uses the shallower feature map in video style transfer task, sacrificing the stylization effects for temporal consistency in some way. MCCNet [7] produces distorted results with severe artifacts along object contours, where the same drawback appears in image style transfer. AdaAttN [25] provides a well-transferred foreground leaving the background requires further rendering. Using the same model as image style transfer, CCPL [38] also shows a huge number of vertical artifacts in video stylization results. In contrast, benefiting from the joint learning framework, our video results are enhanced by the rich texture from images beyond the reference video. Besides, the difference map in column 4 shows that our joint learning framework is stable enough when dealing with the motion blur. Based on the above analysis, our method can generate video results with more pleasing stylistic patterns while maintaining temporal consistency well. Notably, when applied to image style transfer, some of these compared methods [21, 25] need to be retrained with extra changes and consumption, while the rest [7, 38] use the same model directly with the flawed results. In contrast, our joint learning framework can perform well on both image and video style transfer in one go. Additional vivid stylization results are provided in our supplementary materials.

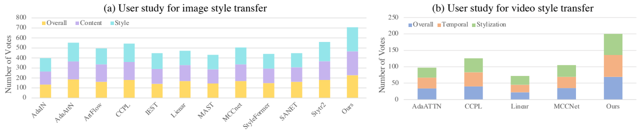


Figure 6: The user study for images (a) and video (b) style transfer. Best viewed by zooming in.

## 4.4. User Study

Furthermore, we carry out the user study experiment for comparison. Specifically, we use 27 content images and 21 style images to synthesize 567 images in total. Given 20 randomly combinations of content and style, the generated results obtained by 12 image-based style transfer methods. Then, we ask 100 participants to select their favorite one from three aspects: content preservation, style transfer, and overall preference. We collect $2,000$ votes and show results in Figure 6(a). The results verifies the superiority of our method over other models for image style transfer. Similarly, we take 12 videos of 50 frames and 21 style images to synthesize 252 video stylized clips in total. Given 4 random combinations of video and style, the stylization clips obtained by 5 video-based style transfer solutions. Then, we ask another 50 subjects to select their favorite one from three views: temporal consistency, stylization effect (considering both content preservation and stylization degree), and overall preference. We collect 200 votes and our method is selected as the best as shown in Figure 6(b).

For participants, there are 83 males and 67 and females (55/45 males/females for image, the other 28/22 males/females for video), aged from 23 to 42.

## 4.5. Efficiency Analysis

In Table 3, we report the inference time of our joint learning method and other approaches. Note that all the methods are run using a single TITAN XP GPU card. Although our joint learning framework consists of many stacked self-attention layers, our model can still achieves 35 FPS at 256px, which is comparable with attention-based methods such as AdaAttN [25] and Stytr$^2$ [9]. The main limitation of our work is the inference speed is limited when applied for high resolution input, which may hinder its usage.

Notably, our AMSA mechanism is crucial for the joint learning framework to address the expensive memory consumption and huge computation complexity. In particular, the computation complexity can be reduced from $\boldsymbol{O(H^2} \times$

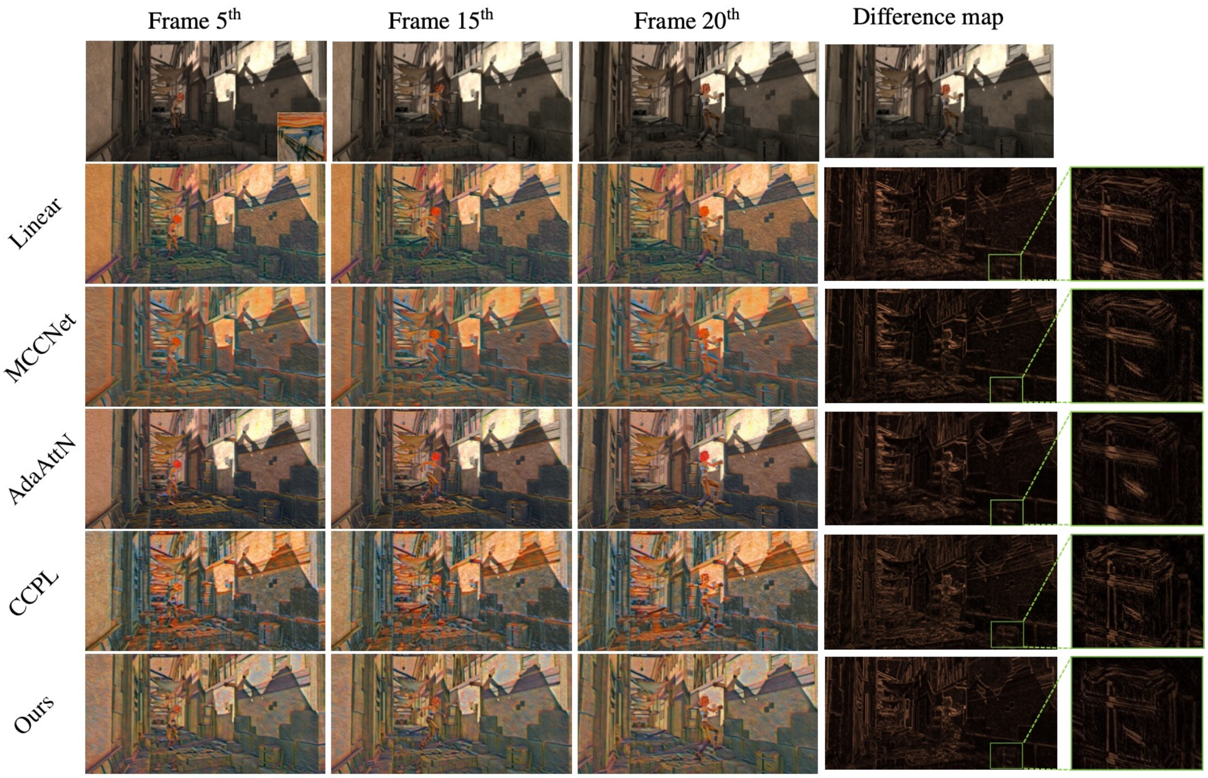


Figure 7: Qualitative comparison in video style transfer. Column 4 shows the difference map between the adjacent frames at frame 20.

| Methods | Ours | SANet | Linear | MCCNet | Artflow | Stytr$^2$ | CCPL |
|---|---|---|---|---|---|---|---|
| $256 \times 256$ | 0.028 | 0.016 | 0.011 | 0.067 | 0.144 | 0.109 | 0.019 |
| $512 \times 512$ | 0.111 | 0.053 | 0.062 | 0.141 | 0.363 | 0.751 | 0.061 |
| Methods | Ours $w/$ MSA | AdaATTN | Avatar-Net | AdaIN | MAST | StyleFormer | IEST |
| $256 \times 256$ | 0.030 | 0.040 | 0.116 | 0.012 | 0.022 | 0.018 | 0.019 |
| $512 \times 512$ | 0.182 | 0.127 | 0.339 | 0.037 | 0.092 | 0.062 | 0.059 |

Table 3: Inference time (sec./image) comparison.

$\boldsymbol{W^2}$) to $\boldsymbol{O(H^2 + W^2)}$ theoretically. As shown in Table 4, AMSA is much lighter compared to MSA under the same experiment setting, demonstrating the great potential for high-resolution image and long video style transfer. Although the framework consists of many self-attention layers, the UniST is still fully capable of practical applications.

| | *Memory(MiB)* | *FLOPS (G)* |
|---|---|---|
| *MSA* | 1.8x10$^4$ | 4.29 |
| *AMSA* | 1.1x10$^4$ | 0.27 |

Table 4: Efficiency of AMSA by comparison with MSA.

### 4.6. Ablation Study

**Video-image interaction module.** Since images and videos have different domain features, we design a cross-attention domain-interaction module for joint learning of two domains for better content knowledge, which improves the overall stylization results. We conduct an experiment by removing the cross-attention interaction module. As in the 2nd, 3rd columns of Table 5, both content preservation and style transfer are highly degraded without this module.

To further demonstrate the superiority of UniST, we compare 3 training strategies without video-image interaction module. As shown in the 4th, 5th, 6th columns of Table 5, UniST achieves the best performance compared with training separately and sequentially. Meanwhile, as shown in Figure 8, video style transfer is improved by learning complex appearances and textures from the image domain

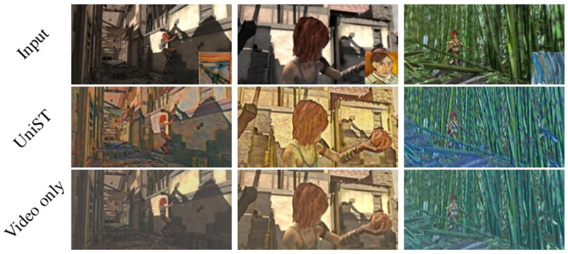


Figure 8: Visualization of video style transfer with different training strategies.

with UniST framework. Similarly, Figure 9 shows UniST enhances image stylization effects with more vivid results.

| | UniST | Image+Video | Video Only | Image Only | Sequential Training |
|---|---|---|---|---|---|
| $\mathcal{D}_C$ (↓) | **12.36** | 15.44 | 14.42 | 19.29 | 16.36 |
| $\mathcal{D}_S$ (↓) | **0.46** | 0.69 | 0.89 | 0.64 | 0.69 |

Table 5: Effectiveness of joint learning framework. “video only” and “image only” indicate unimodal input for training, and “Sequential learning” means that we first train the network with one modality and then fine-tune it with another. Specifically, except for the UniST, the others all use the version without video-image interaction module.

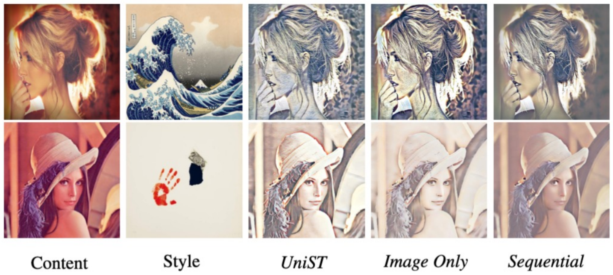


Figure 9: Visualization of image style transfer with different training strategies.

| | Image | Video | Image+Video |
|---|---|---|---|
| $\mathcal{D}_C$(↓) | 12.36 | n/a | 12.36 |
| $\mathcal{D}_S$(↓) | 0.46 | n/a | 0.46 |
| Mean optical flow error(↓) | n/a | 3.86 | 3.86 |
| Mean LPIPS(↓) | n/a | 1.79 | 1.79 |

Table 6: Metrics comparison of different input.

**Inconsistent datasets concerns.** In our method, we adopt two datasets from different modalities for learning. To ensure fairness, we have retrained five image style transfer models, while keeping the training dataset consistent with us. We provide the quantitative comparison in Table 7. Meanwhile, we conduct extra experiments by training the image-only UniST with different dataset scales in Table 8. The results show simply using more images for image-only UniST brings no obvious improvements, while gains come from different modalities and beneficial interactions, and further illustrating the superiority of our model.

| Methods | UniST | CCPL | StyleFormer | IEST | Stytr$^2$ | AdaAttN |
|---|---|---|---|---|---|---|
| $\mathcal{D}_C$(↓) | **12.36** | 14.32 (↓ 0.34) | 13.12 (↓ 3.70) | 15.72 (↑ 0.34) | 12.87 (↓ 0.80) | 14.76 (↑ 0.22) |
| $\mathcal{D}_S$(↓) | **0.46** | 0.85 (↑ 0.14) | 1.13 (↑ 0.33) | 0.97 (↓ 0.41) | 0.61 (↑ 0.11) | 1.24 (↑ 0.17) |

Table 7: Quantitative comparison under consistent training dataset conditions.

**Axial Multi-head Self-Attention.** To verify the effectiveness of AMSA layer, we conduct two different design comparisons in Figure 10. First, the information contained in the previous $K$, $V$ is crucial for the second MSA layer to learn contextual dependencies either within or across domains. Therefore, we uniformly take the output of the first MSA as the $Q$, $K$, $V$ for the second MSA. The results in 4th column show that it is difficult for the model to capture the relevance between content and style domains in this way. Second, the first axial information is necessary for the $V$ of the second MSA to maintain the stable stylization results. To demonstrate this, we use the output of the first MSA as the $Q$ for the second MSA, while keeping the $K$ and $V$ unchanged. As in the 5th column, the style patterns transferred are mixed with bar artifacts. From the results in the 3th column, the original design in Figure 3 is necessary for our model to prevent side effects caused by axial attention.

| Training with different images | $\mathcal{D}_c\downarrow$ | $\mathcal{D}_s\downarrow$ |
|---|---|---|
| UniST with videos and 60K images | 12.36 | 0.46 |
| UniST using videos and 60K images without interaction | 15.44 | 0.69 |
| Image-only UniST with 60K images | 19.80 | 0.73 |
| Image-only UniST with 70K images | 19.80 | 0.76 |
| Image-only UniST with 80K images | 19.98 | 0.71 |

Table 8: Comparison of UniST with image-only UniST using more training images.

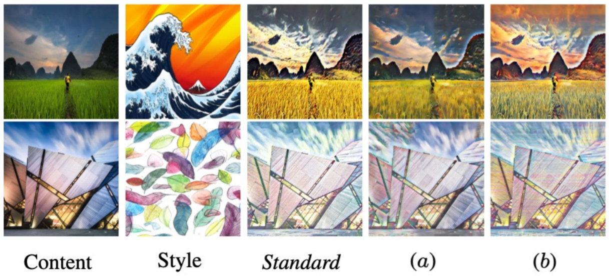


Figure 10: Ablation study of the AMSA layer. “Standard” is the normal version in Figure 3. ($a$) We uniformly take the output of the first MSA as the $Q$, $K$, $V$ of the second MSA. ($b$) We use the output of the first MSA as the $Q$ for the second MSA, while keeping $K$ and $V$ same as the first.

## 5. Conclusion

In this work, we propose an unified style transfer framework, dubbed UniST, for image and video. The key is our novel domain interaction transformer that enables effective mutual feature learning from different modalities for enhancements. Besides, an axial multi-head attention is proposed to capture attentions either within or across the field efficiently. Experiment shows the mixing content input effectively improves stylization results via UniST.

**Acknowledgement.** Libo Zhang was supported by Youth Innovation Promotion Association, CAS (2020111). Heng Fan and his employer received no financial support for research, authorship, and/or publication of this article.

# 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 *CVPR*, pages 862–871, 2021.

[2] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, pages 611–625, 2012.

[3] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent online video style transfer. In *ICCV*, pages 1114–1123, 2017.

[4] Haibo Chen, Lei Zhao, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. Artistic style transfer with internal-external learning and contrastive learning. In *NeurIPS*, pages 26561–26573, 2021.

[5] Jiaxin Cheng, Yue Wu, Ayush Jaiswal, Xu Zhang, Pradeep Natarajan, and Prem Natarajan. User-controllable arbitrary style transfer via entropy regularization. 2023.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[7] Yingying Deng, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, and Changsheng Xu. Arbitrary video style transfer via multi-channel correlation. In *AAAI*, pages 1210–1217, 2021.

[8] Yingying Deng, Fan Tang, Weiming Dong, Wen Sun, Feiyue Huang, and Changsheng Xu. Arbitrary style transfer via multi-adaptation network. In *ACM MM*, pages 2719–2727, 2020.

[9] Yingying Deng, Fan Tang, Xingjia Pan, Weiming Dong, Chongyang Ma, and Changsheng Xu. Stytr^2: Unbiased image style transfer with transformers. *CoRR*, abs/2105.14576, 2021.

[10] Ivan Dokmanic, Reza Parhizkar, Juri Ranieri, and Martin Vetterli. Euclidean distance matrices: Essential theory, algorithms, and applications. *IEEE Signal Process. Mag.*, 32(6):12–30, 2015.

[11] 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. In *ICLR*. OpenReview.net, 2021.

[12] Wei Gao, Yijun Li, Yihang Yin, and Ming-Hsuan Yang. Fast video multi-style transfer. In *WACV*, pages 3211–3219, 2020.

[13] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016.

[14] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *CoRR*, abs/2104.05704, 2021.

[15] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. Real-time neural style transfer for videos. In *CVPR*, pages 7044–7052, 2017.

[16] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, pages 1501–1510, 2017.

[17] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3297–3305. IEEE Computer Society, 2017.

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.

[20] Xiaoyu Kong, Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Yongyong Chen, Zhenyu He, and Changsheng Xu. Exploring the temporal consistency of arbitrary style transfer: A channelwise perspective. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[21] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *CVPR*, pages 3809–3817, 2019.

[22] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *NeurIPS*, 30, 2017.

[23] 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 *CVPR*, pages 5141–5150, 2021.

[24] 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 *ECCV*, pages 740–755, 2014.

[25] 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 *ICCV*, pages 6649–6658, 2021.

[26] Xuan Luo, Zhen Han, Lingkang Yang, and Lingling Zhang. Consistent style transfer. *arXiv preprint arXiv:2201.02233*, 2022.

[27] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *CVPR*, pages 5880–5888, 2019.

[28] Fred Phillips and Brandy Mackintosh. Wiki art gallery, inc.: A case for critical thinking. *Issues in Accounting Education*, 26(3):593–608, 2011.

[29] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *CVPR*, pages 8242–8250, 2018.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 30, 2017.

[32] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Standalone axial-attention for panoptic segmentation. In *ECCV*, pages 108–126, 2020.

[33] Zhizhong Wang, Lei Zhao, Zhiwen Zuo, Ailin Li, Haibo Chen, Wei Xing, and Dongming Lu. Microast: Towards super-fast ultra-resolution arbitrary style transfer. *CoRR*, abs/2211.15313, 2022.

[34] Less Wright. Ranger - a synergistic optimizer. `https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer`, 2019.

[35] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *ICCV*, pages 22–31. IEEE, 2021.

[36] Xinxiao Wu and Jialu Chen. Preserving global and local temporal consistency for arbitrary video style transfer. In *ACM MM*, pages 1791–1799, 2020.

[37] Xiaolei Wu, Zhihao Hu, Lu Sheng, and Dong Xu. Styleformer: Real-time arbitrary style transfer via parametric style composition. In *ICCV*, pages 14598–14607. IEEE, 2021.

[38] Zijie Wu, Zhen Zhu, Junping Du, and Xiang Bai. CCPL: contrastive coherence preserving loss for versatile style transfer. *CoRR*, abs/2207.04808, 2022.

[39] Xide Xia, Tianfan Xue, Wei-Sheng Lai, Zheng Sun, Abby Chang, Brian Kulis, and Jiawen Chen. Real-time localized photorealistic video style transfer. In *WACV*, pages 1088–1097, 2021.

[40] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. *CoRR*, abs/2103.11816, 2021.

[41] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595. Computer Vision Foundation / IEEE Computer Society, 2018.