

# Real-time Localized Photorealistic Video Style Transfer

Xide Xia\*  
 Boston University  
 xidexia@bu.edu

Tianfan Xue  
 Google Research  
 tianfan@google.com

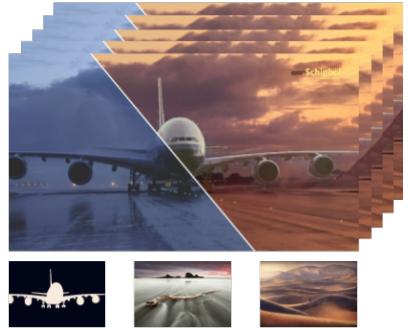
Wei-sheng Lai  
 Google  
 wslai@google.com

Zheng Sun  
 Google Research  
 zhengs@google.com

Abby Chang  
 Google  
 abbychang@google.com

Brian Kulis  
 Boston University  
 bkulis@bu.edu

Jiawen Chen  
 Google Research  
 jiawen@google.com



**Figure 1: Photorealistic video style transfer.** Given an input content video and its object segmentation mask, our method learns to transfer different styles to different local regions while preserving the photorealism and temporal consistency (left: original, right: stylized). The bottom of each example shows the object mask, foreground style image, and background style image, respectively.

## Abstract

We present a novel algorithm for transferring artistic styles of semantically meaningful local regions of an image onto local regions of a target video while preserving its photorealism. Local regions may be selected either fully automatically from an image, through using video segmentation algorithms, or from casual user guidance such as scribbles. Our method, based on a deep neural network architecture inspired by recent work in photorealistic style transfer, is real-time and works on arbitrary inputs without runtime optimization once trained on a diverse dataset of artistic styles. By augmenting our video dataset with noisy semantic labels and jointly optimizing over style, content, mask, and temporal losses, our method can cope with a variety of imperfections in the input and produce temporally coherent videos without visual artifacts. We demonstrate our method on a variety of style images and target videos, including the ability to transfer different styles onto multiple objects simultaneously, and smoothly transition between styles in time.

## 1. Introduction

Color stylization plays a critical role in modern cinematography and video storytelling. It has the powerful ability to grab audience attention, elicit emotions, and convey implicit mood. For example, red usually evokes ideas of action, adventure, and strength; orange can represent joy, creativity and stimulation; and green signifies envy and health. In addition to applying color styles globally to the entire video, modern filmmakers often utilize localized color tone changes; i.e., distinct color palettes applied to certain segmented objects in the scene, as a powerful tool in video storytelling. Some well-known examples include the film *Schindler's List* [14], in which the famous scene of a girl in a red coat becomes the most memorable symbol of the film. Similarly, the film *Sin City* [15] uses high saturation (such as in red and yellow) to colorize certain characters or clothes, contrasting with the film's classic black-and-white noir theme.

While color stylization is critical to video production, color style creation and editing today is still a labor-intensive and time-consuming process even with the use of professional editing software. Many operations are still manual in the editing process, such as rotoscoping, refine-

\* Work done while interning at Google Research.

ment of segmentation masks, ensuring temporal consistency from frame to frame, and carefully fine-tuning color tones. An automatic approach would be incredibly helpful.

To automatically apply color styles to videos, a number of methods have recently been introduced based on advances in deep learning. For instance, several methods have explored the transfer of artistic styles to images or videos [9, 16, 23, 13, 35, 8, 40]. However, due to their nature as *artistic* style transfer methods, they all introduce undesirable painterly spatial distortions. Another line of work [28, 24, 22, 45, 43] focuses on *photorealistic* style transfer, which requires the output to maintain “photorealism”; i.e., the output should look as if it was taken by a real camera (like most stylized films). However, since they primarily target still photography, existing methods often generate visible flicker artifacts when they are applied to videos. To reduce temporal instability, several methods have tried minimizing an optical flow warping loss [12, 11] or sequentially propagating intermediate features [20]. To our knowledge, there is no existing work that can successfully perform photorealistic localized style transfer on videos with reasonably good runtime speed.

In this paper, we present a novel approach that simultaneously addresses three major challenges in localized photorealistic video style transfer: 1) spatial and temporal style coherence over time, 2) robustness against imperfect segmentation masks, and 3) high-speed processing. To achieve high-speed performance, we extend our approach from the recent work of Xia et al. [43], which learns local edge-aware affine transforms from low-resolution content and style inputs, with the results sliced out from a compact transform representation at the full resolution. To minimize spatial artifacts and improve the temporal coherence, we propose a novel spatiotemporal feature transfer layer (ST-AdaIN) that is able to transfer style to local regions and generate temporally coherent stylized videos. To handle imperfect segmentation masks, our algorithm learns an enhancement network to improve the boundaries of masks. Moreover, we compose foreground and background color transform coefficients in the grid space, resulting in high quality results even if segmentation masks are inaccurate. Experimental results demonstrate that our model generates stylized videos with fewer visual artifacts and better temporal coherence than existing photorealistic style transfer methods.

In summary, our contributions in this work are threefold:

- A differentiable spatiotemporal style transfer layer (ST-AdaIN) to match the feature statistics of local regions, generating temporally coherent stylized results (Section 3.2).
- Mask enhancement and grid-space blending algorithms that can render natural style transitions between objects given noisy selection masks (Section 3.3).
- A deep neural network for photorealistic video style

transfer that runs in real-time (26.5 Hz at  $1024 \times 1024$  resolution).

## 2. Related Work

**Image style transfer.** Classical style transfer approaches can be categorized into: 1) global methods, which match global image statistics [32, 30], and 2) local methods [18, 36, 37, 42, 41], which find dense correspondence between content and style. While the global methods are efficient, the results are not always faithful to the style image. The local methods can generate high-quality results, but they are computationally expensive and often limited to specific scenarios (e.g., portraits, sky, or season changes).

Convolutional neural network (CNN)-based style transfer has been widely studied in recent years. The pioneering work of Gatys et al. [9] formulates the problem as an iterative optimization to match the statistics of feature activations within a pre-trained classification network. Several follow-up work either improves stylization efficiency by learning strictly feed-forward networks [16, 21], or increase generalization by adding the ability to transfer multiple styles within a single “universal” model [13, 35, 23]. Although these approaches produce impressive artistic stylization results, the images often contain spatial distortions and warped image structures that are unacceptable in a photorealistic setting.

To achieve photorealistic style transfer, Luan et al. [28] extend the optimization framework of Gatys et al. [9] by imposing a local affine *photorealism constraint*. PhotoWCT [24] optimizes the loss functions of Luan et al. [28] in a closed-form solution but requires a post-processing step to further smooth the stylized results. Li et al. [22] learn linear transformations but require a spatial propagation network [25] as an anti-distortion filter. On the other hand, WCT<sup>2</sup> [45] adopts the wavelet corrected transfer based WCT to preserve the photorealism without any post-processing. An et al. [2] adopt neural architecture search and network pruning to learn a lightweight model, but their model is still behind real-time performance. Recently, Xia et al. [43] propose a model that strictly enforces Luan’s photorealism constraint by directly learning style transfer in a bilateral space [10], which can achieve real-time performance at 4K resolution. The proposed model builds upon this fast method but is able to transfer styles to local regions and generate temporally coherent video results.

**Video style transfer.** As style transfer methods typically generate new texture or change image color significantly, applying existing algorithms to videos frame-by-frame often lead to temporally inconsistent results. To improve the temporal stability, several methods minimize a temporal loss (i.e., optical flow warping error [3]) in an iterative optimization framework [34], training feed-forward net-

works [12, 11], or learning a post-processing module [20]. Chen et al. [4] propagate long-range information by blending the intermediate features with the ones from the previous frame. Our method enforces short-temporal consistency by minimizing the temporal loss and imposes long-term temporal consistency by propagating the intermediate features of the neighboring frames through the proposed spatiotemporal feature transfer layer.

**Bilateral grid.** Bilateral space is commonly used for fast image processing. Paris and Durand [29] first introduce bilateral space for fast edge-aware image filtering, showing that it can be sampled at a low resolution *bilateral grid*. Chen et al. [5] extend the bilateral grid for fast approximation of many edge-aware image transformations. Recently, Gharbi et al. [10] learn a deep network to predict bilateral grid for approximating several image transformations. The grid is only estimated from a low-resolution input and then applied to the full-resolution input image, significantly accelerating the speed of complicated image transformation. Xia et al. [43] further extend this structure to photorealistic image style transfer by integrating with the Adaptive Instance Normalization (AdaIN) [13]. In this work, we also propose to model the style transformation using bilateral grids, and we will describe the technical differences and improvement against Xia et al [43] in Section 3.

### 3. Method

We aim to transfer the style from a set of images to different local regions in a target video, producing a smooth, temporally-consistent, and photorealistic result. We take as input a video (the *content*), object selection masks (one video per object), and a set of style images (each style corresponds to one object), and output a video sequence. Since manual rotoscoping to produce selection masks is tedious, and automatic object detection is rarely pixel-perfect, one critical design goal in localized video style transfer is the ability to handle imperfect and noisy masks. Without loss of generality, we discuss the case of having two regions that we call foreground (i.e., where the mask value is 1) and background (i.e., the complement). Extending the algorithm to three or more regions is straightforward with linear complexity.

#### 3.1. Overview

Our method builds upon the style transfer algorithm of Xia et al. [43], which we summarize here. The algorithm consists of two “paths” (Figure 2), a full-resolution frame processing path (in purple) and a low-resolution grid prediction path (in green). The grid prediction path takes as input a low-resolution content/style pair (e.g.,  $256 \times 256$ ), extracts features using a pre-trained VGG-19 network [38], and uses a series of Adaptive Instance Normalization (AdaIN) lay-

ers [13] to predict a photorealistic style transformation operator. The transformation operator is encoded as a very low-resolution affine bilateral grid  $\Gamma$  (e.g.,  $16 \times 16 \times 8$ ). To render a full-resolution frame in real-time, the full-resolution path uses a per-pixel learned lookup table to first predict a *guide map*. The guide map serve as a learned proxy for image luminance that better separates edges. To render a given input pixel at position  $(x, y)$  and color  $(r, g, b)$ , the algorithm computes its guide value  $z = LUT(r, g, b)$ , slices the affine bilateral grid  $\Gamma$  at  $(x, y, z)$  (sampling with trilinear interpolation and scaling each axis) to retrieve a  $3 \times 4$  affine transform  $A$ , and computes the product  $A \cdot (r, g, b, 1)^T$ . Our work is motivated by the fact that while Xia et al.’s algorithm is fast and works well on still photos, our early experiments showed that it is nontrivial to extend the method to localized video style transfer. A naive localized extension would be to simply run the algorithm twice on each frame: once per region, and then blend using object masks. However, this does not work well due to the following reasons:

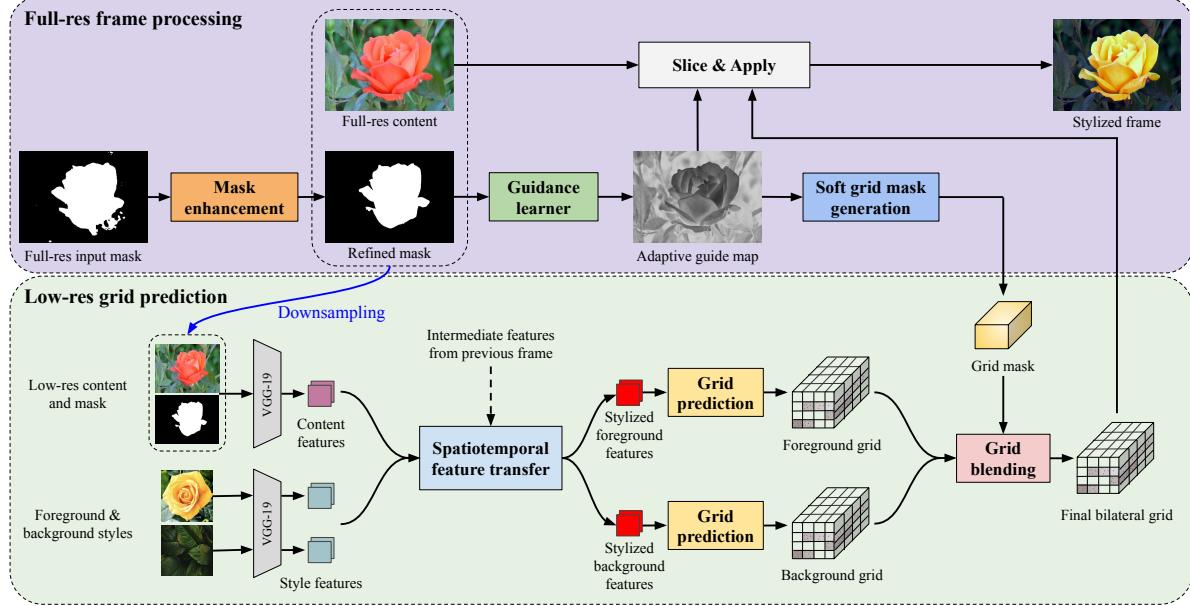
- Content statistics upon which stylization operates need to be localized and temporally coherent to produce a satisfactory output. This requires a new feature transfer operator (Section 3.2).
- Pixel-space blending using object masks depends critically on mask quality, which is rarely satisfactory. We address this by learning to enhance masks (Section 3.3), which are then incorporated into the rendering process (Section 3.4).
- Although Xia et al. [43] show a few smooth video results despite being trained on only still imagery, we find that without additional temporal regularization, there is noticeable flicker. We address this with the proposed spatiotemporal feature transfer layer and a warping loss (Section 3.2 and 3.5).

#### 3.2. Spatiotemporal Feature Transfer

Recall that AdaIN [13] transfers the mean  $\mu(\cdot)$  and variance  $\sigma(\cdot)$  feature statistics from  $y$  (e.g., style) onto  $x$  (e.g., content) as:

$$\text{AdaIN}(x, y) = \sigma(y) \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y). \quad (1)$$

Mean and variance are computed independently for each feature channel. Note that the original AdaIN operator computes mean and variance over an entire frame. This is inappropriate for localized style transfer as it mixes the features of foreground and background regions, leading to visible halo artifacts near object boundaries (see Figure 6(d) for an example). We address this with a *spatially-aware AdaIN* (SA-AdaIN), which extends the operator to account for the



**Figure 2: Model architecture.** Our model consists of two paths. The *grid prediction* path consumes low-resolution content, style, and masks and infers a bilateral grid that encodes how to transform each layer. At full resolution, we predict a guide map that tells us how to best blend the style transforms, as well as how to index into it for rendering. See Section 3 for details.

content mask  $m_x$ :

$$\text{SA-AdaIN}(x, y, m_x) = \sigma(y) \left( \frac{x - \mu(x \cdot m_x)}{\sigma(x \cdot m_x)} \right) + \mu(y), \quad (2)$$

Next, we tackle motion. Although using AdaIN on VGG-19 features has shown to be effective for still photo style transfer [13, 43], it has been shown that these features are sensitive to small changes in the input. Subtle motions can lead to large differences in the features [11] which are amplified by downstream stylization that appears as flicker artifacts when viewed at framerate [20]. To address this, we propose a *temporally coherent AdaIN* (TC-AdaIN) layer that computes the mean and variance from both the previous and current frames:

$$\text{TC-AdaIN}(x^t, x^{t-1}, y, \alpha) = \sigma(y) \left( \frac{x - \mu^t}{\sigma^t} \right) + \mu(y), \quad (3)$$

where  $\mu^{(t)} = (1 - \alpha)\mu(x^{(t)}) + \alpha\mu(x^{(t-1)})$ ,  $\sigma^{(t)} = (1 - \alpha)\sigma(x^{(t)}) + \alpha\sigma(x^{(t-1)})$ , and  $\alpha$  is a weight that balances the contribution of the previous ( $t - 1$ ) and current ( $t$ ) frame.

Combining (2) and (3), we obtain a *spatiotemporal AdaIN* (ST-AdaIN) operator:

$$\text{ST-AdaIN}(x^t, x^{t-1}, y, m_x^t, m_x^{t-1}) = \sigma(y) \left( \frac{x - \hat{\mu}^t}{\hat{\sigma}^t} \right) + \mu(y), \quad (4)$$

where

$$\hat{\mu}^t = (1 - \alpha)\mu(x^t \cdot m_x^t) + \alpha\mu(x^{t-1} \cdot m_x^{t-1}), \quad (5)$$

$$\hat{\sigma}^t = (1 - \alpha)\sigma(x^t \cdot m_x^t) + \alpha\sigma(x^{t-1} \cdot m_x^{t-1}). \quad (6)$$

After the feature transferring, we apply a few convolutional layers to predict the bilateral grids for both the foreground and background. The detailed architectures of our spatiotemporal feature transfer and grid prediction are provided in the supplementary material.

### 3.3. Mask Enhancement

In practice, input selection masks are rarely pixel-perfect, either due to errors made by object segmentation algorithms (e.g., [26, 6, 7]) or inaccuracies in users' annotations. To ensure that masking errors are not propagated to our stylized output, we introduce a mask enhancement network that smooths and aligns masks to object boundaries. Our mask enhancement network is a lightweight module, consisting of three convolutional layers, where the last layer adopts a sigmoid activation to produce a soft boundary.

### 3.4. Stylized Rendering

After learning bilateral grids that encode foreground and background styles, we need to smoothly blend between them to produce a photorealistic output. A naive solution would be to apply the grids independently to the input frame to produce two stylized outputs, and alpha blend them using the enhanced mask calculated in Section 3.3. However,

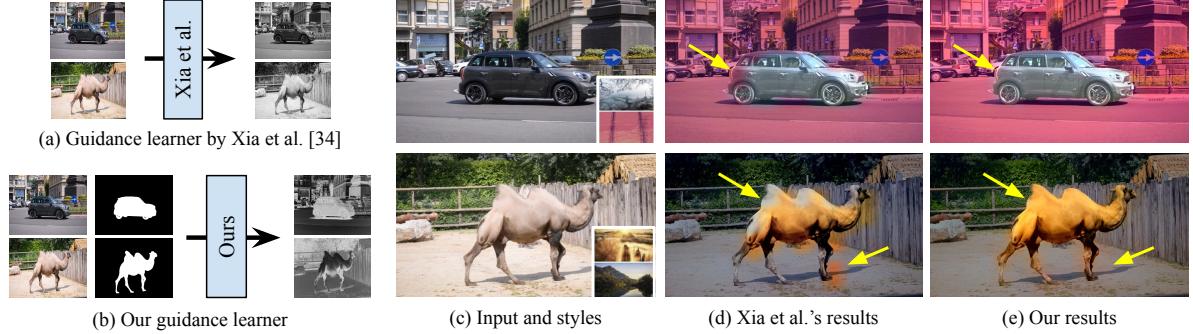


Figure 3: **Guide map for grid blending and slicing.** While Xia et al.[43] learns the guidance from the input image alone, our model takes the object mask as an extra input, which allows it to better separate foreground and background at boundaries.

applying the transformation twice at full-resolution is not only inefficient, but if the two styles (and thus transformations) are dramatically different, it will also lead to undesirable halo artifacts (see Figure 3(d)) despite blending using a soft mask. We devise a better solution that interchanges the steps: first *blend the transforms* taking masks into account, then apply the blended transform once at full resolution.

Recall that HDRnet [10] and bilateral photo style transfer [43] learn a global lookup table based on pixel color alone to produce its guide map. The guide map, which is directly used as the  $z$  index when slicing the bilateral grid, serves to separate object edges. This is insufficient in our application since object edges are delineated not only by color gradients of large magnitude, but also by explicit masks. Therefore, we replace the learned lookup table with a 2-layer convolutional neural network that also incorporates the object mask to learn an adaptive guide map that better separates foreground and background, as shown in Figure 3(a).

We use the adaptive guide map to build a soft *grid-space* mask (see the supplementary material for details) and combine the two stylization transforms into a single compact representation as:

$$\Gamma = M_{grid} \cdot \Gamma_f + (1 - M_{grid}) \cdot \Gamma_b, \quad (7)$$

where  $\Gamma_f$  and  $\Gamma_g$  are the foreground and background grids, respectively. Finally, we render the output at each pixel by slicing out an affine transform from blended grid  $\Gamma$  using the adaptive guide map and performing a matrix multiply. Figure 3(c) and (d) shows how grid-space blending eliminates artifacts from the naive approach.

### 3.5. Training Losses

Given a collection of input content videos  $\{I_c\}$  with object masks, foreground and background style images  $F_s$  and  $B_s$ , our model can be trained to generate a stylized output video  $\{O\}$  on given an arbitrary input by optimizing the

following multi-objective loss function:

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s + \lambda_r \mathcal{L}_r + \lambda_m \mathcal{L}_m + \lambda_z \mathcal{L}_z + \lambda_t \mathcal{L}_t. \quad (8)$$

We empirically set  $\lambda_c = 0.2$ ,  $\lambda_s = 1$ ,  $\lambda_r = 0.02$ ,  $\lambda_m = 5.0$ ,  $\lambda_z = 1.5$ , and  $\lambda_t = 1000$  in all experiments. We discuss each component below.

We adopt the now standard *content loss* that minimizes the difference between the intermediate features from a pre-trained VGG-19 network [9]:

$$\mathcal{L}_c = \sum_{i=1}^{N_C} \|\Phi_i[O] - \Phi_i[I_c]\|_F^2, \quad (9)$$

where  $\Phi_i[\cdot]$  is a feature map,  $N_C$  denotes the number of layers selected from VGG-19 to build content loss, and  $\|\cdot\|_F$  is the Frobenius norm. Our *style loss* uses the simpler formulation from recent work [13, 43] that uses only the mean and variance of of VGG-19 features:

$$\begin{aligned} \mathcal{L}_s &= \sum_{i=1}^{N_S} \|\mu(\Phi_i[O]) - \mu(\Phi_i[I_s])\|_2^2 \\ &+ \sum_{i=1}^{N_S} \|\sigma(\Phi_i[O]) - \sigma(\Phi_i[I_s])\|_2^2, \end{aligned} \quad (10)$$

where  $N_s$  denotes the number of layers selected from VGG-19 to build style loss.

To penalize abrupt local changes in the predicted transforms but allow changes across strong object edges (known as *bilateral smoothness*), we adopt the *Laplacian regularizer* of [43] on both the foreground grid  $\Gamma_f$  and background grids  $\Gamma_b$ :

$$\begin{aligned} \mathcal{L}_r &= \sum_s \sum_{t \in N(s)} \|\Gamma_f[s] - \Gamma_f[t]\|_F^2 \\ &+ \sum_s \sum_{t \in N(s)} \|\Gamma_b[s] - \Gamma_b[t]\|_F^2. \end{aligned} \quad (11)$$

where  $N(s)$  denotes the neighbors of cell  $s$ .

**Mask loss.** To ensure that the network does not predict an affine transformation that applies the foreground style to a background pixel and vice versa, we add a *mask loss* that says: if we sliced out a full-resolution pixel-space mask from the predicted grid mask using our learned guide map, it should align with object boundaries. Mathematically:

$$\mathcal{L}_m = \|slice(z, M_{grid}) - M_{gt}\|_2^2, \quad (12)$$

where  $z$  is the learned guide map,  $M_{grid}$  is the predicted soft grid mask, and  $M_{gt}$  is the ground-truth object mask.

**Guide loss.** We penalize guide maps that are substantially different from input luminance  $I_{gray}$  since luminance edges are a strong segmentation signal:

$$\mathcal{L}_z = \|z - I_{gray}\|_2^2. \quad (13)$$

**Temporal loss.** To improve temporal coherence, we minimize the flow warping error between the output frames [20]:

$$\mathcal{L}_t = \sum_{t=1}^T \sum_{i=1}^N V_{t \rightarrow t-1}^i \|O_t^i - \hat{O}_{t-1}^i\|_1, \quad (14)$$

where  $\hat{O}_{t-1}$  is frame  $O_{t-1}$  warped by optical flow  $f_{t \rightarrow t-1}$  and  $V_{t \rightarrow t-1}$  is the visibility mask calculated from the input frames  $I_t$  and  $I_{t-1}$ . We use the PWC-Net [39] to compute optical flow on the content video. More implementation details are provided in the supplementary material.

## 4. Experimental Results

In this section, we first provide qualitative comparisons with state-of-the-art photorealistic style transfer methods. We then present quantitative results from our performance benchmark and user study. Finally, we analyze the contribution of different components in our model.

### 4.1. Setup

We use the DAVIS 2017 validation set [31] for evaluation, which contains 30 videos with ground-truth segmentation masks and the video resolution is  $480 \times 584$ . In our experiments, we adopt an instance segmentation method, COSNet [27], to select a single instance as foreground. Our system can also take multiple instances as foreground with additional user input (e.g., “control click”).

We compare our method to state-of-the-art photorealistic style transfer algorithms: PhotoWCT [24]<sup>1</sup>, LST [22], WCT<sup>2</sup> [45], and Xia et al. [43]. Note that we apply the full pipeline of PhotoWCT and LST, including their

<sup>1</sup>We use NVIDIA’s latest FastPhotoStyle library, which is much faster than the speed claimed in their paper.

own smoothing steps. Since these approaches are intended for still image style transfer, to make the comparison more fair, we apply the temporal consistency algorithm of Lai et al. [20] to as a post-process to improve their results. As the method of Xia et al. [43] cannot stylize local regions, we extend it by first generating two stylized outputs for foreground and background, then blending them in pixel-space using the input mask.

### 4.2. Qualitative Comparisons

Figure 4 provides visual comparisons on two examples. As both PhotoWCT and LST transfer style features on the intermediate layers of a deep neural network with an encoder-decoder architecture, foreground and background styles are mixed together. Hence, the final output synthesized by the decoder contains halo artifacts at object boundaries (e.g., the swan’s neck for PhotoWCT) or spatially inconsistent color (e.g., the uneven grass). Those visual artifacts cannot be easily removed by their post-processing steps. Therefore, the results of PhotoWCT and LST still look non-photorealistic with noticeable artifacts. While WCT<sup>2</sup> and Xia et al. [43] do not require any post-processing, their stylized results contain visual artifacts around the object boundaries due to inaccuracies in the input object masks. In contrast, our method produces visually pleasing results given imperfect segmentation masks. By using our mask-enhancement and grid-space blending, we can spatially blend between substantially different styles without explicitly separating foreground and background with pixel precision. We provide video comparisons in the supplementary material, which shows that our model generates results that are more temporally stable.

### 4.3. Quantitative Results

**Performance.** We measure execution time on a single NVIDIA Tesla V100 GPU with 16 GB of RAM at various resolutions. For existing methods [22, 24, 45, 43], we include time spent on post-processing using Lai’s method [20] (also listed independently). As shown in Table 1, our method is much faster than all but the method of Xia et al. [43] upon which it is based. This is due to our model’s additional steps of computing a soft grid-space mask. Still, our model can run at 21.5 Hz on 12 megapixel inputs, while other approaches run out of memory. We note that Xia et al. [43] alone can run on  $3000 \times 4000$  input, but the post-processing by Lai et al. [20] runs out of memory at this resolution.

**User study.** As style transfer fidelity is inherently a subjective matter of taste, we conduct a user study to evaluate user’s preferences. We adopt paired comparisons [33, 19], where users are asked to choose the better result in a pair shown side-by-side. In each test, we also provide the input content video, object segmentation mask, foreground, and

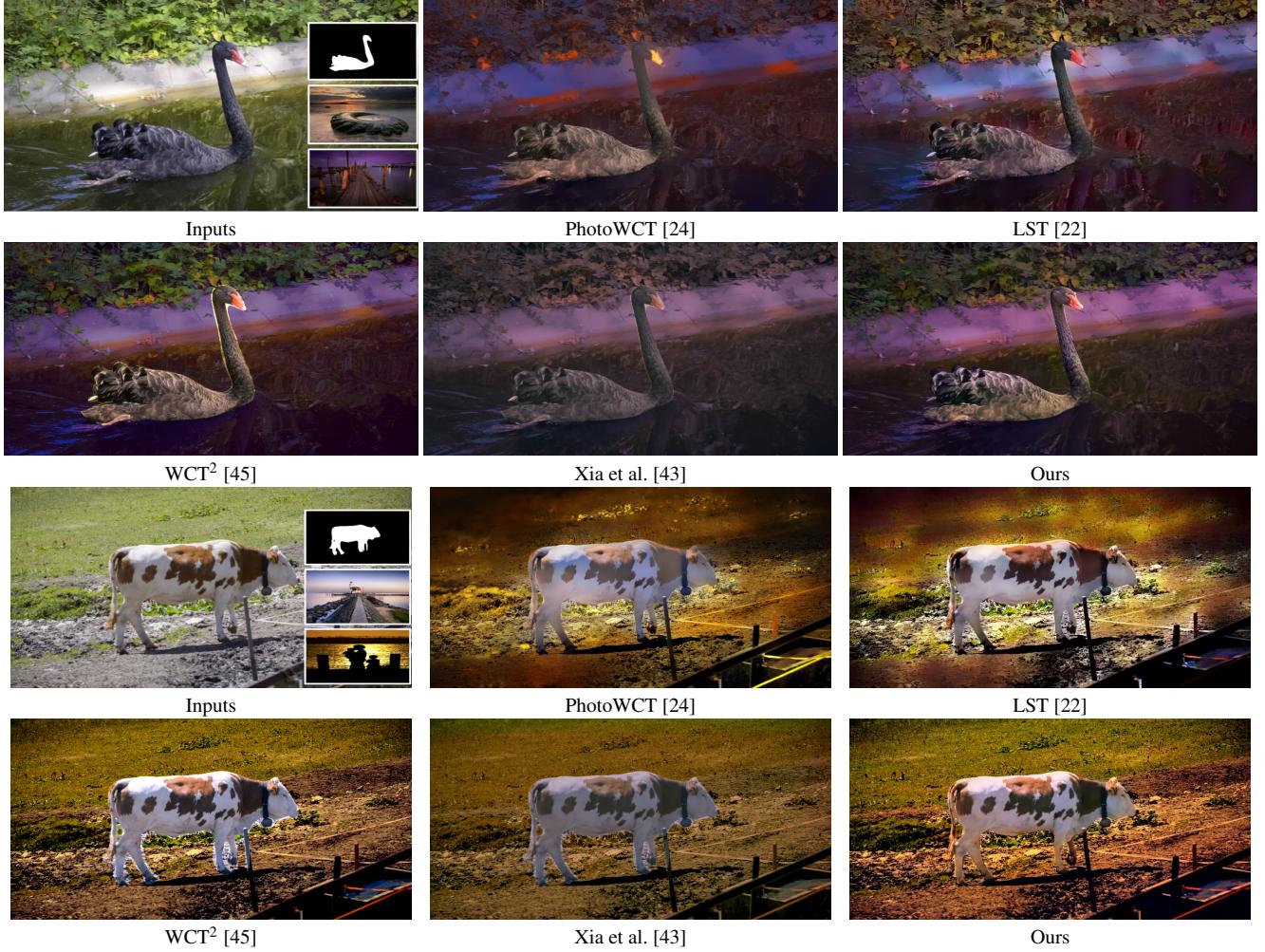


Figure 4: **Visual comparisons.** The proposed method generates more visually pleasing stylized results than state-of-the-art photorealistic style transfer algorithms.

| Image Size             | $512 \times 512$    | $1024 \times 1024$  | $2000 \times 2000$  | $3000 \times 4000$  |
|------------------------|---------------------|---------------------|---------------------|---------------------|
| Lai et al. [20]        | $0.002 \pm 0.001$ s | $0.003 \pm 0.001$ s | $0.008 \pm 0.003$ s | OOM                 |
| LST* [22]              | $0.260 \pm 0.125$ s | $0.793 \pm 0.334$ s | OOM                 | OOM                 |
| PhotoWCT* [24]         | $0.624 \pm 0.298$ s | $1.548 \pm 0.359$ s | OOM                 | OOM                 |
| WCT <sup>2*</sup> [45] | $3.904 \pm 0.280$ s | $6.203 \pm 0.453$ s | OOM                 | OOM                 |
| Xia et al.* [43]       | $0.005 \pm 0.001$ s | $0.007 \pm 0.002$ s | $0.013 \pm 0.004$ s | OOM                 |
| Ours                   | $0.039 \pm 0.001$ s | $0.040 \pm 0.012$ s | $0.043 \pm 0.013$ s | $0.046 \pm 0.014$ s |

Table 1: **Execution time.** An asterisk (\*) denotes that the technique of Lai et al. [20] is applied. OOM indicates out of memory at the inference time.

background style images as reference. The participant is asked to answer the following questions:

1. Which video has a more faithful stylization?
2. Which video has fewer visual artifacts?
3. Which video is more temporally stable?

In total, we recruit 30 participants, where each participant evaluates 20 sets of videos. While the results are shuffled randomly, we ensure that all the methods are compared the

|                                 | Better stylization | Fewer visual artifacts | Better temporal stability |
|---------------------------------|--------------------|------------------------|---------------------------|
| Ours vs. LST* [22]              | $60.6 \pm 7.5\%$   | $68.5 \pm 7.2\%$       | $66.1 \pm 7.3\%$          |
| Ours vs. PhotoWCT* [24]         | $53.9 \pm 7.7\%$   | $80.6 \pm 6.1\%$       | $73.9 \pm 6.8\%$          |
| Ours vs. WCT <sup>2*</sup> [45] | $57.6 \pm 7.6\%$   | $79.4 \pm 6.2\%$       | $73.9 \pm 6.8\%$          |
| Ours vs. Xia et al.* [43]       | $69.1 \pm 7.1\%$   | $60.6 \pm 7.5\%$       | $57.6 \pm 7.6\%$          |
| Average                         | $60.3 \pm 3.7\%$   | $72.3 \pm 3.4\%$       | $67.9 \pm 3.6\%$          |

Table 2: **Results of user study.** An asterisk (\*) denotes that the technique of Lai et al. [20] is applied. Overall, our results are preferred by more than 60% of users.

same number of times.

Table 2 shows the percentage that our result is preferred over the other. Overall, our method is selected on more than 60% of comparisons for all three questions, demonstrating that our results have a more faithful stylization, fewer visual artifacts, and better temporal stability.



Figure 5: **Style transition.** Our model can smoothly transit between styles in time by interpolating the bilateral grids.

#### 4.4. Style Transition

The style and color tone in a film can change over time, connoting a gradual shift in mood. Our method can easily transition between different styles in time by interpolating between predicted grids using an appropriate time-dependent weighting function (e.g., a user-selected spline). Figure 5 samples a few frames from a result where we transfer different styles onto foreground and background while smoothly transitioning in time. The full video result is shown in the supplementary file.

#### 4.5. Ablation Studies

**Grid-space blending and SA-AdaIN.** As input object masks are often noisy, blending in pixel space often results in visible artifacts around object boundaries (Figure 6(c)). To render natural object boundaries, one may need to apply image matting [44] to feather the mask. But matting is both computationally expensive and may introduce its own artifacts. In contrast, our method learns how to blend the foreground and background style transfer transforms, sidestepping the need for an accurate full-resolution mask (Figure 6(e)). Furthermore, by adopting the SA-AdaIN (Equation (2)), we improve the separation of foreground and background styles better and avoid the classic halo artifacts as shown in Figure 6(d).

**Temporal consistency.** To quantitatively evaluate the temporal consistency of the stylized videos, we measure the warping error (14) and temporal change consistency (TCC) [46]. We compare the per-frame results of PhotoWCT [24], LST [22], WCT<sup>2</sup> [45], and Xia et al. [43], both with and without applying the temporal smoothing technique of Lai et al. [20]. We also train two modified versions of our model: one without the temporal loss and one without TC-AdaIN. As shown in Table 3, our full model achieves the lowest warping error and the highest TCC, demonstrating that our method achieves the most stable results. Figure 7 also shows a visual comparison, where the results without the TC-AdaIN have a visible color shift.

### 5. Conclusion

In this work, we propose a novel algorithm for photorealistic video style transfer. Our algorithm is able to 1) generate spatially and temporally coherent stylized results through the proposed spatiotemporal feature trans-

| Model                                   | Warping error ( $\downarrow$ ) | TCC ( $\uparrow$ ) |
|---|--------------------------------|--------------------|
| PhotoWCT [24]                           | 0.00124                        | 0.632              |
| LST [22]                                | 0.00129                        | 0.534              |
| WCT <sup>2</sup> [45]                   | 0.00118                        | 0.566              |
| Xia et al. [43]                         | 0.00098                        | 0.607              |
| PhotoWCT [24] + Lai et al. [20]         | 0.00099                        | 0.643              |
| LST [22] + Lai et al. [20]              | 0.00104                        | 0.538              |
| WCT <sup>2</sup> [45] + Lai et al. [20] | 0.00110                        | 0.576              |
| Xia et al. [43] + Lai et al. [20]       | 0.00093                        | 0.660              |
| Ours w/o $\mathcal{L}_t$                | 0.00091                        | 0.673              |
| Ours w/o TC-AdaIN                       | 0.00094                        | 0.659              |
| <b>Ours</b>                             | <b>0.00090</b>                 | <b>0.688</b>       |

Table 3: **Temporal consistency evaluation.** Our full model achieves the lowest warping error and the highest TCC against the existing algorithms and our own variations.

fer layer, 2) account for imperfect input masks by learning a mask enhancement network and blending the color transform coefficients in a low-resolution grid space, and 3) achieve real-time performance by adopting an efficient bilateral learning framework. We demonstrate the performance of the proposed method on a wide variety of videos and styles. Finally, a human subject study shows that our method achieves faithful stylization, higher visual quality, and better temporal coherence against existing photorealistic style transfer approaches.

## 6. Algorithm Details

### 6.1. Network architectures

We provide the architectural details of our mask enhancement network, guidance learner, spatiotemporal feature transfer, and grid learner in Table 4. As shown in Figure 8, we extract the VGG-19 features from four scales (CONV1\_1, CONV2\_1, CONV3\_1, and CONV4\_1) and match the multi-resolution feature statistics via three splatting blocks [43]. Note we remove the global scene summary path in Xia et al. [43] as we aim to transfer style between local regions, where the global features will not help. In each splatting block, the weights of the first and the last convolutional layers are shared for the content and style feature paths. The output of each ST-AdaIN layer is propagated to the next frame for blending the temporal information. After the feature transferring, we apply two convolutional layers to predict the bilateral grids.

### 6.2. Soft grid mask

In Algorithm 1, we provide the detailed steps to compute the soft grid mask for blending the foreground and background grids.

### 6.3. Implementation details

We implement our model in Tensorflow [1]. For training, we use the training set from DAVIS 2017 [31] which con-



Figure 6: **Effect of grid-space blending and SA-AdaIN.** Blending in the pixel-space results in unnatural object boundaries due to the hard edge in the mask. Our grid-space blending can render smoother boundaries, but the foreground style bleeds to the background (the yellow arrows in (d)). With the proposed SA-AdaIN, our model can separate the foreground and background styles well and render high-quality object boundaries.

|              | Mask Enhancement |       |         | Guidance Learner |         | Spatiotemporal Feature Transfer |         |         |         |         |         |         |         |         |       |       |      | Grid Prediction |  |
|--------------|------------------|-------|---------|------------------|---------|---------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|-------|-------|------|-----------------|--|
|              | $M^1$            | $M^2$ | $M^3$   | $G^1$            | $G^2$   | $S_1^1$                         | $S_1^2$ | $S_1^3$ | $S_2^1$ | $S_2^2$ | $S_2^3$ | $S_3^1$ | $S_3^2$ | $S_3^3$ | $L^1$ | $L^2$ | $F$  | $\Gamma$        |  |
| type         | conv             | conv  | conv    | conv             | conv    | conv                            | conv    | conv    | conv    | conv    | conv    | conv    | conv    | conv    | conv  | conv  | conv | conv            |  |
| stride       | 1                | 1     | 1       | 1                | 1       | 2                               | 1       | 1       | 2       | 1       | 1       | 2       | 1       | 1       | 1     | 1     | 1    | 1               |  |
| kernel size  | 3                | 3     | 3       | 3                | 3       | 3                               | 3       | 3       | 3       | 3       | 3       | 3       | 3       | 3       | 3     | 3     | 3    | 1               |  |
| spatial size | 256              | 256   | 256     | 256              | 256     | 128                             | 128     | 128     | 64      | 64      | 64      | 32      | 32      | 32      | 16    | 16    | 16   | 16              |  |
| channels     | 16               | 8     | 1       | 16               | 1       | 8                               | 8       | 8       | 16      | 16      | 16      | 32      | 32      | 32      | 64    | 64    | 64   | 96              |  |
| activation   | -                | -     | sigmoid | -                | sigmoid | relu                            | relu    | relu    | relu    | relu    | relu    | relu    | relu    | relu    | relu  | relu  | relu | -               |  |

Table 4: Detailed configuration of each module.



Figure 7: **Effect of TC-AdaIN.** Without propagating intermediate features with TC-AdaIN, the stylized outputs have a significant color shifting within 8 frames in this example.

tains 60 videos and each video has 70 frames in average. We optimize the model using Adam [17] with hyperparameters  $\alpha = 10^{-4}$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 10^{-8}$ , and a batch size of 2 video clips. At each iteration, we take five consecutive frames as a clip and randomly crop the original frames into a resolution of  $256 \times 256$ . To train the mask enhancement network, we generate noisy input masks by applying erosion and dilation with random kernel sizes to the ground-truth object masks. Since mask enhancement is a largely orthogonal task, we first pre-train this module for 20000 iterations and freeze the weights. We then train the remainder of the network end-to-end for 90000 iterations. By limiting the “full-res” input to also be  $256 \times 256$ ,

#### Algorithm 1: Compute Soft Grid Mask

---

**Input:** Learned guide map  $\mathbf{z}$ , pixel mask  $\mathbf{M}_{\text{pxl}}$ , image size  $(w, h)$ , grid size  $(W, H, D)$

**Output:** Soft grid mask  $\mathbf{M}_{\text{grid}}$

- 1 Initialize grid mask:  $M_{\text{grid}} = \text{zeros}(W, H, D)$ ;
- 2  $\mathbf{z}_D = \text{floor}((\mathbf{z} \cdot \mathbf{M}_{\text{pxl}}) \times D)$  ;
- 3  $s_w, s_h = w/W, h/H$  ;
- 4 **for**  $x \leftarrow 1$  to  $W$  and  $y \leftarrow 1$  to  $H$  **do**
- 5      $\mathbf{patch} = \mathbf{z}_D[x \times s_w : (x+1) \times s_w, h \times s_h : (y+1) \times y_h]$  ;
- 6      $M_{\text{grid}}[x, y, :] \leftarrow \text{sum}(\mathbf{patch} > 0)$ ;
- 7     **for**  $d \leftarrow 1$  to  $D$  **do**
- 8         **if**  $d$  in  $\mathbf{patch}$  **then**
- 9              $M_{\text{grid}}[w, h, d] \leftarrow \text{sum}(\mathbf{patch} == d)$ ;
- 10         **end**
- 11     **end**
- 12 **end**
- 13 Normalize grid mask:  $M_{\text{grid}} \leftarrow M_{\text{grid}} / (s_w \times s_h)$ ;

---

we can significantly reduce training time, which takes about two days on a single NVIDIA Tesla V100 GPU with 16 GB RAM. At inference time, the trained model can be applied to arbitrary input resolution, because of the usage of bilateral grid.

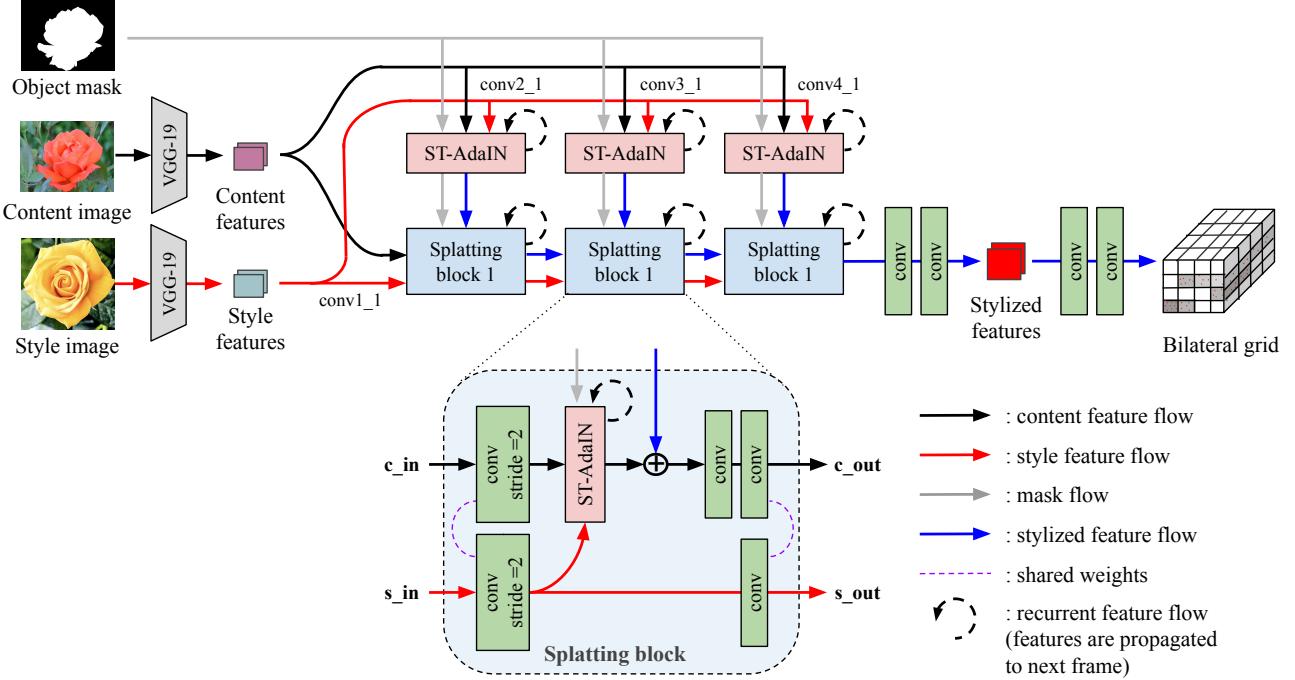


Figure 8: Detailed architecture of our spatiotemporal feature transfer and grid prediction layers.

## 7. Grid Sub-sampling

The proposed method can be further sped up by subsampling the bilateral grids in the temporal domain. As the nearby frames typically have similar color, the bilateral grids can also be shared to render the stylized result. Specifically, we can generate the bilateral grid for every  $r$  frames and estimate the intermediate grids by a linear interpolation, as shown in Figure 9(a). In Figure 9(c) and (d), we show that our method can still render high-quality results up to a sub-sampling rate of  $r = 8$ . For  $r = 16$ , the estimated grid may have a larger spatial mismatch with the content, resulting in undesired visual artifacts. Such a temporal sub-sampling strategy is suitable for the proposed method and Xia et al. [43]. Other approaches, e.g., WCT<sup>2</sup> [45], is not able to generate reasonable results by simply interpolating the output frames, as shown in Figure 9(b).

By utilizing such a grid sub-sampling strategy, we can reduce the computational cost and speed up the processing time during inference. As shown in Table 5, our model with sub-sampling rate  $r = 8$  is about  $8 \times$  faster at four different image resolutions. The video comparisons are provided in the supplementary videos *grid\_sampling.mp4* in the **video\_mp4** folder.

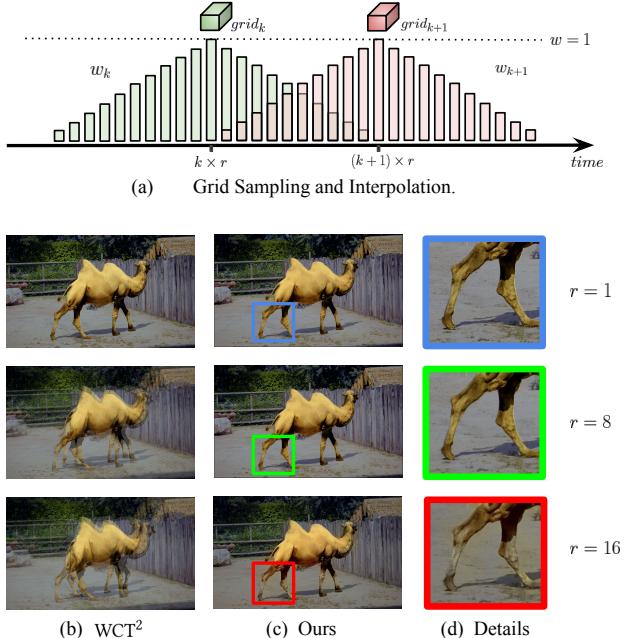


Figure 9: Our method can be sped up by subsampling the bilateral grids in the temporal space while achieving similar visual quality.

| Image Size              | $512 \times 512$ | $1024 \times 1024$ | $2000 \times 2000$ | $3000 \times 4000$ |
|-------------------------|------------------|--------------------|--------------------|--------------------|
| Lai et al. [20]         | 0.0024s          | 0.0031s            | 0.0073s            | OOM                |
| LST* [22]               | 0.2753s          | 0.8365s            | OOM                | OOM                |
| PhotoWCT* [24]          | 0.6366s          | 1.5185s            | OOM                | OOM                |
| WCT <sup>2</sup> * [45] | 3.8599s          | 6.1375s            | OOM                | OOM                |
| Xia et al.* [43]        | 0.0058s          | 0.0068s            | 0.0117s            | OOM                |
| Ours ( $r = 1$ )        | 0.0378s          | 0.0380s            | 0.0414s            | 0.0464s            |
| Ours ( $r = 8$ )        | 0.0048s          | 0.0049s            | 0.0052s            | 0.0058s            |

Table 5: Execution time. An asterisk denotes that the technique of Lai et al. [20] was applied to improving temporal stability. OOM indicates out of memory.

## 8. More Qualitative Comparisons

### 8.1. Mask refinement

While our grid-space blending can handle imperfect input masks, visible artifacts may remain if the masks are too noisy, as shown in Figure 10(b) and (e). Our mask enhancement network significantly improves mask boundaries (Figure 10(c)), and our rendered result (Figure 10(f)) is visually comparable to a rendering using the ground truth mask (Figure 10(g)).

### 8.2. Anti-distortion module

Recent photorealistic style transfer methods [24, 22, 45] are based on encoder-decoder architecture, which often cause spatial distortions or unrealistic visual artifacts when reconstructing an image from the low-resolution deep features. Therefore, extra smoothing steps or modules are required to minimize those spatial distortions. PhotoWCT [24] optimizes a matting affinity to ensure spatially consistent stylization. Similarly, LST [22] applies a spatial propagation network (SPN) [25] on the reconstructed images to smooth the results. On the other hand, WCT<sup>2</sup> [45] replaced max-pooling layers with wavelet pooling where the high frequency components are skipped to the decoder directly so that all edges and corners are preserved.

Here, we show visual comparisons to existing methods with and without applying their anti-distortion modules. Figure 11 shows the comparison with PhotoWCT. Although the smoothing step helps reduce local artifacts, the smoothed result becomes blurry and hazy. In contrast, our result preserves all the edges and image structures well. In Figure 12, the result of LST without applying SPN has severe spatial distortions. The SPN recovers some image details but cannot suppress all the distortions, resulting in an unrealistic result. Figure 13 shows the results of WCT<sup>2</sup> with and without the skip connections for high-frequency components. It is clear that the skip connections help bring back the image details and preserve the photorealism, showing comparable results to the proposed method. More video comparisons are provided in the supplementary videos *visual\_comparison.mp4* in the **video\_mp4** folder.

### 8.3. High-resolution style transfer

Although the proposed method is efficient when processing high-resolution (e.g.,  $2000 \times 2000$ ) videos, extracting the object segmentation masks could be computationally expensive. For example, it takes 2.61 seconds for a segmentation model, DeepLab [7], to process a single  $1920 \times 1080$  frame. Therefore, we aim to understand the feasibility of applying low-resolution segmentation masks for high-resolution style transfer. First, we compute the object masks from a low-resolution ( $256 \times 256$ ) input frame, which takes only 0.47 seconds for DeepLab. Then, we resize the masks to  $1920 \times 1080$  through nearest neighbor interpolation for transferring styles. We compare the proposed method with WCT<sup>2</sup> in Figure 14. The results from WCT<sup>2</sup> have clear visual artifacts on the object boundaries as the upscaled masks are noisy. On the other hand, our method is able to generate high-quality results, which are comparable to the ones produced by using the ground-truth object masks. We demonstrate that our model performs well on high-resolution videos even if the segmentation masks are extracted from a low-resolution space. Please refer to the supplementary videos *HD\_video\_w\_upscaled\_mask\_\*.mp4* in the **video\_mp4** folder for video comparisons.

## 9. Failure Cases

Our method requires reliable image-space statistics and can cope with modest amounts of segmentation noise. However, it may fail when the selected areas are too small (e.g., a textureless region with a single color). On the other hand, when the foreground objects cannot be detected properly due to occlusion or reflection, our method may not be able to separate the styles of foreground and background very well, as shown in Figure 15.

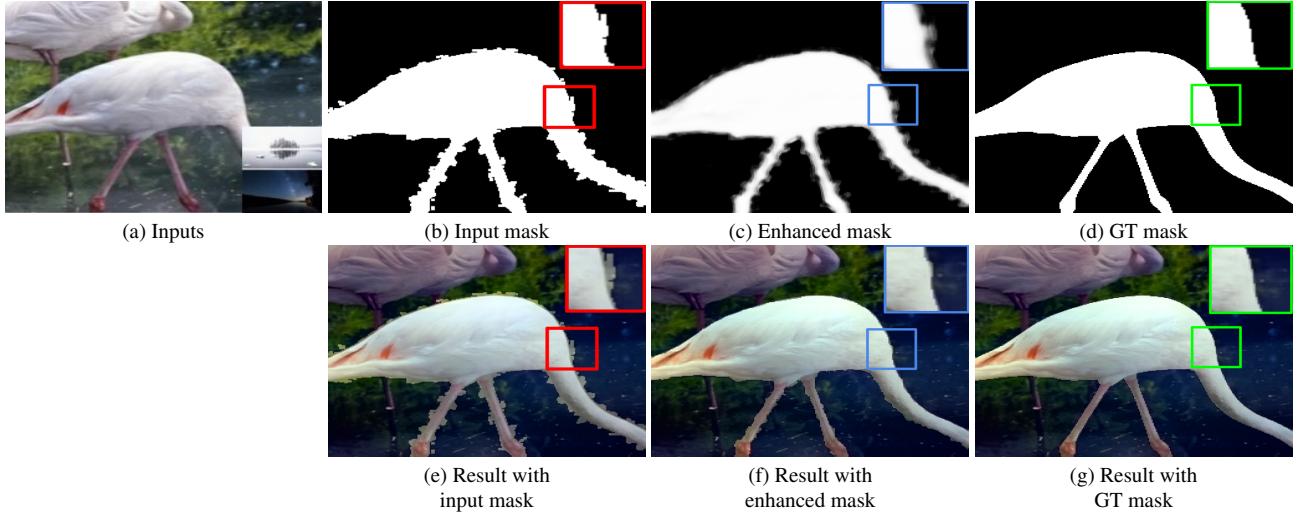


Figure 10: Effect of mask enhancement. When the input object mask is too noisy, the stylized results may still contain visible artifacts on the boundaries. Mask enhancement lets us render high-quality boundaries visually comparable to the result using the ground truth mask.

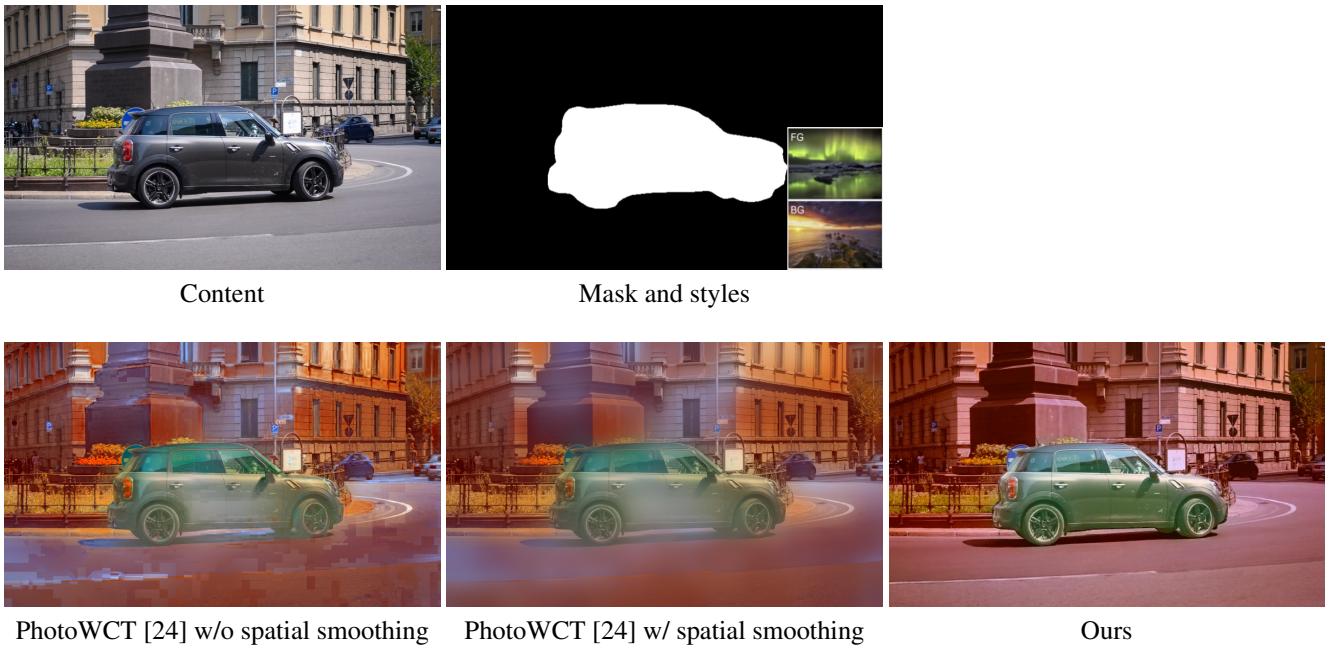


Figure 11: Visual comparison with PhotoWCT [24].

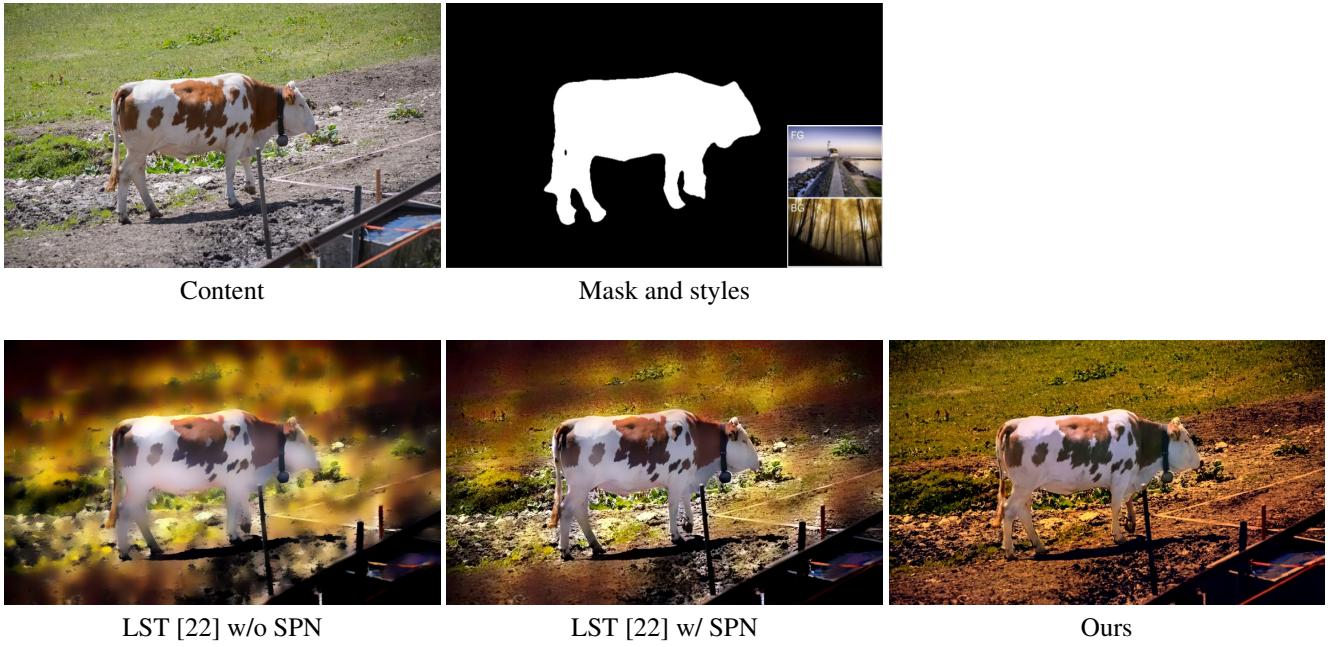


Figure 12: Visual comparison with LST [22].

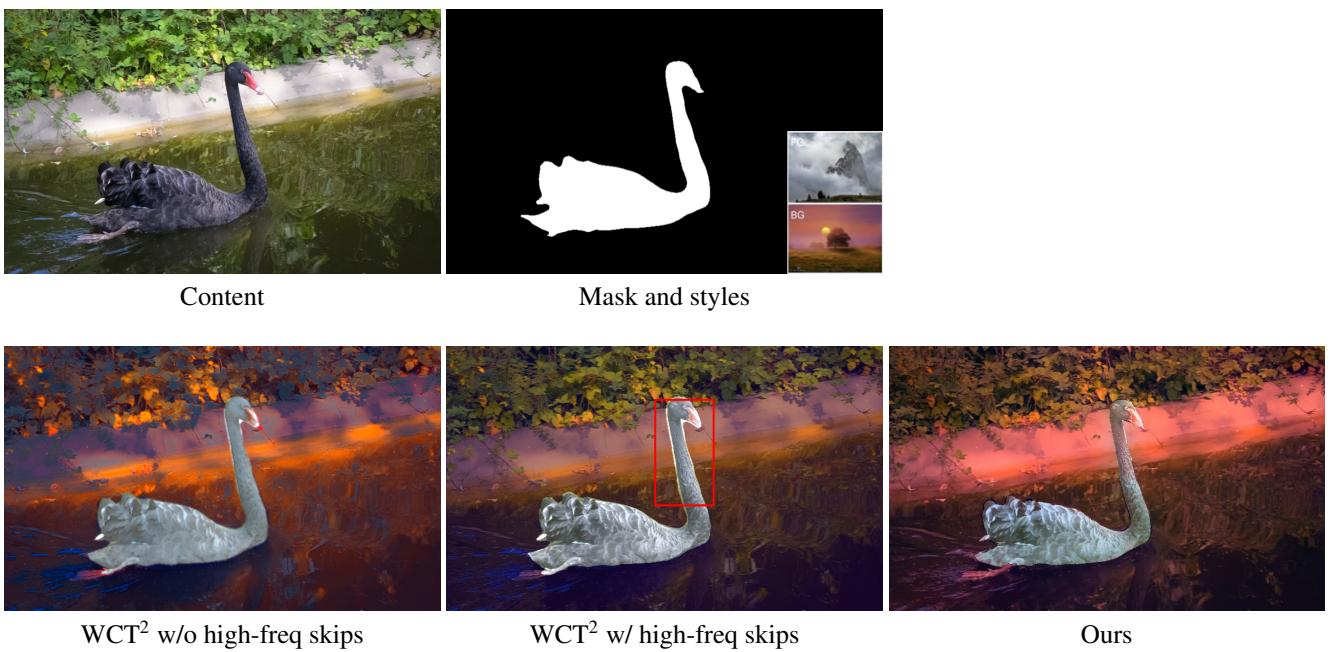


Figure 13: Visual comparison with WCT<sup>2</sup>.

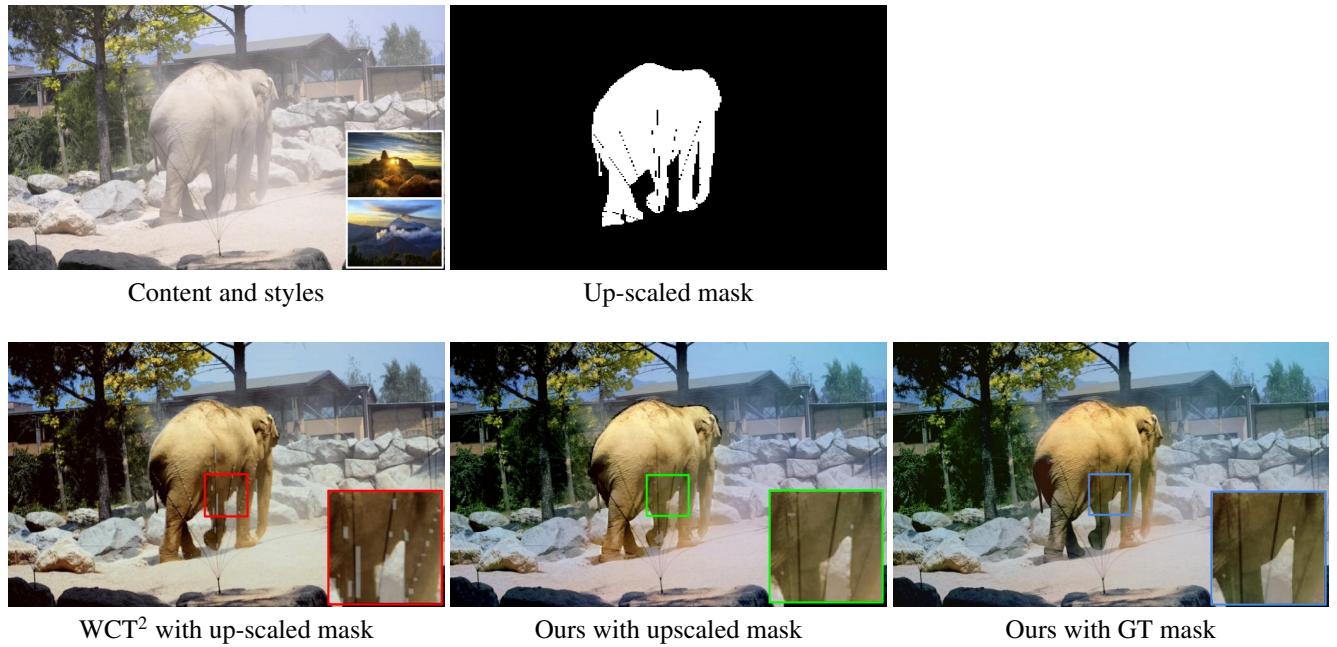


Figure 14: High-resolution video stylization using low-resolution masks.

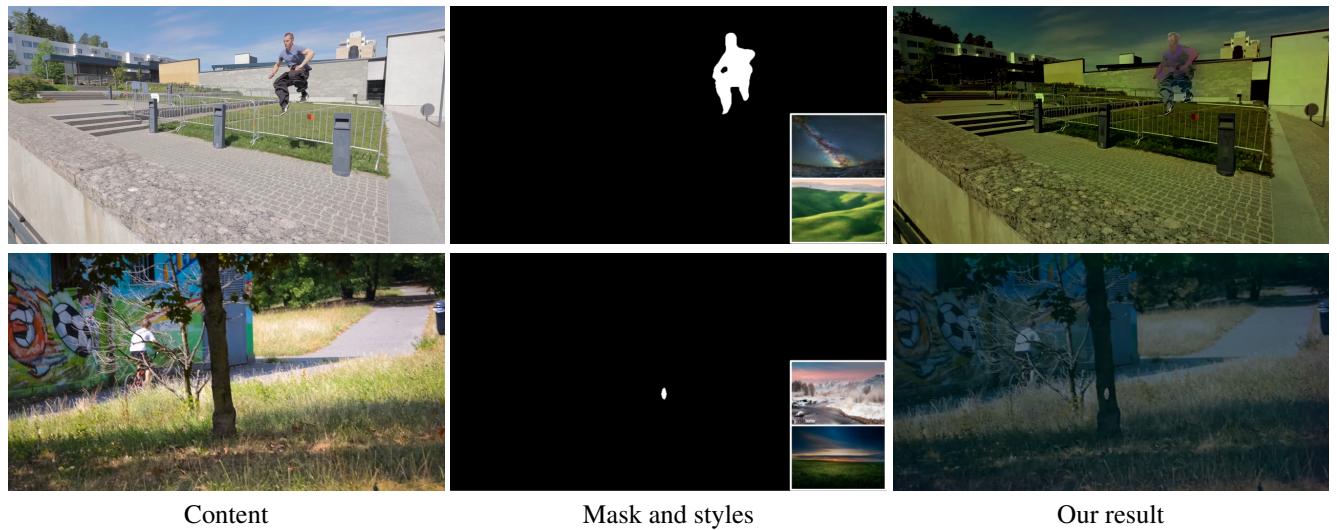


Figure 15: Failure cases.

## References

- [1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016.
- [2] Jie An, Haoyi Xiong, Jun Huan, and Jiebo Luo. Ultrafast photorealistic style transfer via neural architecture search. In *AAAI*, 2020.
- [3] Nicolas Bonneel, James Tompkin, Kalyan Sunkavalli, Deqing Sun, Sylvain Paris, and Hanspeter Pfister. Blind video temporal consistency. *ACM TOG*, 34(6):1–9, 2015.
- [4] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent online video style transfer. In *CVPR*, 2017.
- [5] Jiawen Chen, Andrew Adams, Neal Wadhwa, and Samuel W Hasinoff. Bilateral guided upsampling. *ACM TOG*, 35(6):203, 2016.
- [6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2017.
- [7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.
- [8] Xinghao Chen, Yiman Zhang, Yunhe Wang, Han Shu, Chunjing Xu, and Chang Xu. Optical flow distillation: Towards efficient and stable video style transfer. In *ECCV*, 2020.
- [9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.
- [10] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédéric Durand. Deep bilateral learning for real-time image enhancement. *ACM TOG*, 36(4):118, 2017.
- [11] Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Characterizing and improving stability in neural style transfer. In *ICCV*, 2017.
- [12] 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*, 2017.
- [13] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.
- [14] IMDB. Schindler’s list, 1993.
- [15] IMDB. Sin city, 2005.
- [16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.
- [17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [18] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM TOG*, 33(4):1–11, 2014.
- [19] Wei-Sheng Lai, Jia-Bin Huang, Zhe Hu, Narendra Ahuja, and Ming-Hsuan Yang. A comparative study for single image blind deblurring. In *CVPR*, 2016.
- [20] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018.
- [21] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, 2016.
- [22] Xuetong Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *CVPR*, 2019.
- [23] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *NIPS*, 2017.
- [24] Yijun Li, Ming-Yu Liu, Xuetong Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *ECCV*, 2018.
- [25] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *NIPS*, 2017.
- [26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.
- [27] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *CVPR*, 2019.
- [28] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *CVPR*, 2017.
- [29] Sylvain Paris and Frédéric Durand. A fast approximation of the bilateral filter using a signal processing approach. In *ECCV*, 2006.
- [30] François Fleuret, Anil C Kokaram, and Rozenn Dahyot. N-dimensional probability density function transfer and its application to color transfer. In *ICCV*, 2005.
- [31] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbelaez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv:1704.00675*, 2017.
- [32] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(5):34–41, 2001.
- [33] Michael Rubinstein, Diego Gutierrez, Olga Sorkine, and Ariel Shamir. A comparative study of image retargeting. In *ACM SIGGRAPH Asia*, 2010.
- [34] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *German Conference on Pattern Recognition*, 2016.
- [35] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *CVPR*, 2018.
- [36] YiChang Shih, Sylvain Paris, Connelly Barnes, William T Freeman, and Frédéric Durand. Style transfer for headshot portraits. *ACM TOG*, 33(4):148, 2014.
- [37] Yichang Shih, Sylvain Paris, Frédéric Durand, and William T Freeman. Data-driven hallucination of different times of day from a single outdoor photo. *ACM TOG*, 32(6):1–11, 2013.

- [38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [39] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.
- [40] Ondřej Texler, David Futschik, Michal Kučera, Ondřej Jamriška, Šárka Sochorová, Menglei Chai, Sergey Tulyakov, and Daniel Sýkora. Interactive video stylization using few-shot patch-based training. *ACM TOG*, 39(4):73, 2020.
- [41] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, and Ming-Hsuan Yang. Sky is not the limit: semantic-aware sky replacement. *ACM TOG*, 35(4):149–1, 2016.
- [42] Fuzhang Wu, Weiming Dong, Yan Kong, Xing Mei, Jean-Claude Paul, and Xiaopeng Zhang. Content-based colour transfer. *Computer Graphics Forum*, 32(1):190–203, 2013.
- [43] Xide Xia, Meng Zhang, Tianfan Xue, Zheng Sun, Hui Fang, Brian Kulis, and Jiawen Chen. Joint bilateral learning for real-time universal photorealistic style transfer. In *ECCV*, 2020.
- [44] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *CVPR*, 2017.
- [45] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *ICCV*, 2019.
- [46] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation. In *CVPR*, 2019.