Line Search-Based Feature Transformation for Fast, Stable, and Tunable Content-Style Control in Photorealistic Style Transfer

Tai-Yin Chiu
University of Texas at Austin
chiu.taiyin@utexas.edu

Danna Gurari
University of Colorado Boulder
Danna.Gurari@colorado.edu

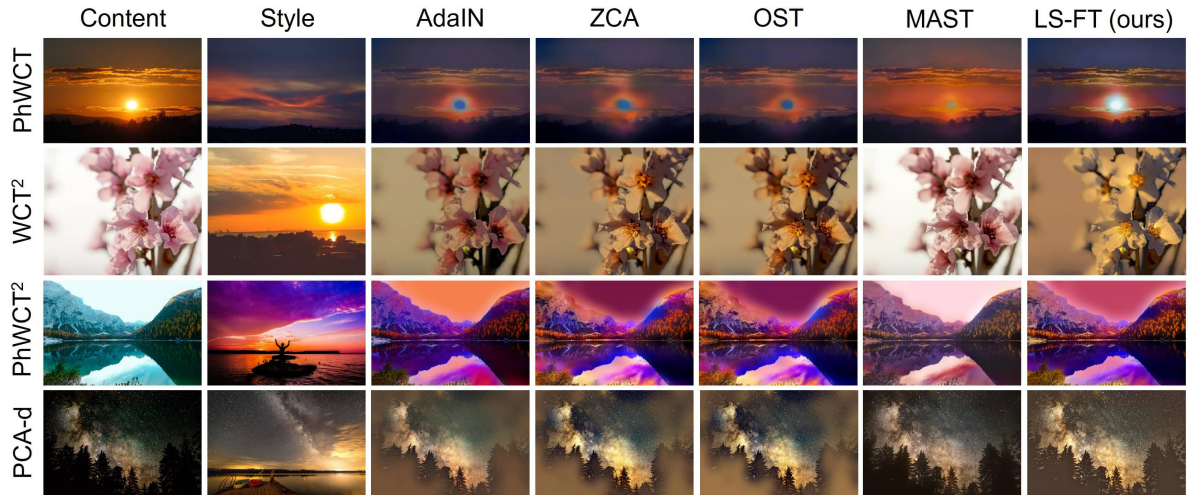

Figure 1: Results are shown for embedding five transformations in four different autoencoder-based photorealistic style transfer models: WCT^2 [31], PhotoWCT (PhWCT) [16], PhotoWCT^2 (PhWCT^2) [6], and a distilled model (PCA-d) [7]. Our new transformation leads to a better balance between content preservation and style transfer than existing transformations. Compared to AdaIN [10], ZCA [15], OST [20], and MAST [11], our LS-FT can preserve better content with PhotoWCT and PCA-d, boost the stylization strength of WCT^2, and reach a better content-style balance with PhotoWCT^2.

Abstract

Photorealistic style transfer is the task of synthesizing a realistic-looking image when adapting the content from one image to appear in the style of another image. Modern models commonly embed a transformation that fuses features describing the content image and style image and then decodes the resulting feature into a stylized image. We introduce a general-purpose transformation that enables controlling the balance between how much content is preserved and the strength of the infused style. We offer the first experiments that demonstrate the performance of existing transformations across different style transfer models, and demonstrate how our transformation performs better in its ability to simultaneously run fast, produce consistently reasonable results, and control the balance between content and style in different models. To support reproducing our method and models, we share the code at https://github.com/chiutaiyin/LS-FT.

1. Introduction

Photorealistic style transfer is an image editing task that renders a content image with the style of another image, which we call the style image, such that the result looks like a realistic photograph to people. In this paper, we tackle a key challenge of how to preserve the content while ensuring strong adoption of the style.

The state-of-the-art approaches for photorealistic style transfer consist of an autoencoder with feature transformations [16, 31, 1, 6, 7]. Such approaches are advantageous in that they can transfer style from any arbitrary style image (i.e., are universal), are fast since they predict in a single forward pass, and do not involve training on any style images (i.e., are style-learning-free). The basic model contains an encoder to extract the features of the content and style images, a feature transformation to adapt the content

feature with respect to the style feature, and a decoder to convert the adapted feature to the stylized image (exemplified in Fig. 2(a)). More advanced models embed multiple transformations to achieve better aesthetic results (e.g., PhotoWCT^2 [6] exemplified in Fig. 2(b)).

A limitation of advanced models is the lack of a general-purpose, stable, one-size-fits-all transformation that yields content-style balance across all models, as exemplified qualitatively in Fig. 1. The most commonly used transformations in photorealistic style transfer models are AdaIN [10] and ZCA [15]. Yet, both AdaIN and ZCA can fail to faithfully reflect the style of style images when embedded in WCT^2 and do not preserve content well when embedded in PhotoWCT (Fig. 1). When embedded in PhotoWCT^2, AdaIN can suffer insufficient stylization strength while ZCA can introduce artifacts that ruin the photorealism. When embedded in PCA-d, ZCA and AdaIN can lead to severe artifacts, as will be shown in Sec. 4.2.

Recently, an iterative feature transformation [5] (IterFT) was proposed which has a control knob to tune between content preservation and style transferal and so can adaptively address the limitation of content-style imbalance across different models. However, as will be explained in Sec. 3.1 and Sec. 3.2, this transformation in practice is unstable as it often produces poor results. Additionally, it is relatively slow.

In this work, we provide several contributions. We expose the problem that existing transformations do not generalize well when used with different photorealistic style transfer models through extensive experiments. We address the limitations of existing transformations by introducing a new transformation, which we call LS-FT. Our experiments show that it consistently achieves a better balance of content preservation and stylization strength by permitting tuning between these two competing aims. Additionally, our experiments show that it runs 8x faster and consistently produces more reasonable results than the only other transformation that can tune between content preservation and stylization strength: IterFT. Ablation studies reveal the key mechanisms behind our transformation's improved performance: introduction of two steps to IterFT (centralization and decentralization) and line-search based optimization.

2. Related Works

Photorealistic style transfer models. DPST [21] is the first deep neural network-based model for photorealistic style transfer. However, DPST is slow since it needs hundreds of times of feed-forward and back-propagation to render an image. To solve this issue, most modern photorealistic style transfer models use autoencoders to render an image in a single forward pass.[1]

[1] An exception is [30] which learns affine mappings to alter pixel values.

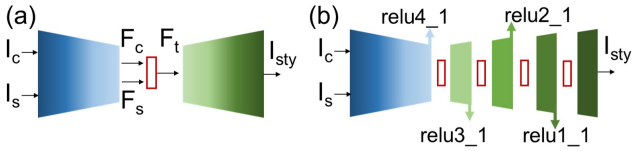

Figure 2: Autoencoder-based algorithms for photorealistic style transfer. (a) The basic model with a feature transformation (red box) at the bottleneck takes as input a content image I_c and a style image I_s and produces a stylized image I_{sty}. (b) PhotoWCT^2 [6] with multiple transformations embedded in the decoder sequentially adapts the *relu4_1* to the *relu1_1* content features.

Some of these autoencoder models [14, 13, 3, 9] are trained on style images. Yet, these have not emerged as state-of-the-art for a variety of reasons. For instance, some only work for low resolution input images, such as DTP [13] which supports only 256×256 resolution. Others often produce unrealistic results, as exemplified for LST [14] and DSTN [9] in the Supplementary Materials.

The other models are style-learning-free autoencoders. To achieve stylization, they embed transformations, which are the focus of our work. These models come with numerous advantages including the flexibility to embed different transformations to achieve different purposes, such as fast speed, strong stylization strength, and better photorealism. Style-learning-free models include PhotoWCT [16], WCT^2 [31], PhotoNAS [1], PhotoWCT^2 [6], and PCA-d [7]. However, a lingering challenge lies in knowing what transformations to embed in style transfer architectures. For example, many popular models use ZCA as the feature transformation, yet prior work [6] has shown it leads to weaker style effects for WCT^2 and PhotoNAS.

We introduce the first study where we pair popular transformations with multiple style transfer architectures.[2] We demonstrate limitations of existing transformations and show that our new transformation, LS-FT, generalizes better due to its ability to balance content preservation and stylization strength while also running fast.

Feature transforms for photorealistic style transfer. Recently, numerous transformations have been proposed. AdaIN [10] is the simplest, which adapts the content feature to match the standard deviation vector and the mean vector of the style feature. Extending AdaIN, ZCA [15, 4] considers the cross correlation between channels and transforms the content feature to match the mean and covariance of the style feature. Experimentally [15, 16, 1], ZCA results in stronger stylization strength than AdaIN. OST [20] further modifies the covariance matching operation of ZCA to better preserve the content while maintaining stronger

[2] PhotoNAS is excluded from the main paper since its model size is too large to even handle the small HD image resolution on the GPU.

	AdaIN	ZCA	OST	MAST	IterFT	LS-FT
Fast	✓	✓	✓	✗	✗	✓
Consistent Results	✓	✓	✓	✓	✗	✓
Content-Style Control	✗	✗	✗	✗	✓	✓

Table 1: Comparison of our LS-FT to existing transformations. Our LS-FT is the only one that simultaneously realizes three beneficial properties.

stylization strength. However, the improvement in content preservation is limited. A limitation across all these transformations is they lack a way to control the balance between content preservation and style transferal.

To address the limitations of prior transformations, iterative feature transformation (IterFT) [5] was proposed. Its main advantage over its prior work is that it supports tuning between content preservation and style transferal to meet the needs of each model. Yet, our experiments show this transformation often produces unreasonable results. Additionally, it is relatively slow. To overcome these shortcomings of IterFT while taking advantage of its ability to be tuned to the needs of different models, we extend it and propose a new transformation dubbed LS-FT. Our experiments demonstrate that LS-FT not only realizes model adaptiveness in practice, but consistently produces reasonable results while performing 8x faster than IterFT.

Of note, a recently proposed transformation MAST [11] takes a different approach from the aforementioned transformations and our LS-FT. Specifically, while most transformations adapt the content feature as a whole, MAST adapts each content feature pixel with respect to the style feature pixels that are semantically close. Yet, as shown in Fig. 1, MAST can only weakly stylize content images.

We summarize the properties of each transformation in Tab. 1, demonstrated by our experiments. Our transformation improves upon prior work by simultaneously being fast (e.g., faster or comparable speed to ZCA in Tab. 2), rendering consistently good results (Fig. 3 and Fig. 9 in the Supplementary Materials), and achieving content-style control (Fig. 5 and Figs. 6-7 in the Supplementary Materials).

Image translation. Image translation [2, 18, 24, 23, 12, 19, 26, 22] models are trained to render images from one domain in the style of another domain (ex: day $\rightarrow$ night) and so are able to adapt image style. Unlike photorealistic style transfer, image translation is not universal meaning that we need to retrain the models when the domains of interest change. Image translation goes beyond style transfer by also altering content (ex: straight hair $\rightarrow$ curly hair).

3. Method

We now introduce our feature transformation that we designed for general-purpose use across multiple style transfer architectures. We begin by describing in Section 3.1 the iterative feature transformation (IterFT) we redesign. Then we introduce the two key ingredients that enable the strengths of our transformation compared to IterFT: a modification that leads to consistently high quality results (Sec. 3.2) and a *line search-based feature transformation* (LS-FT) which enables faster speeds (Sec. 3.3).

3.1. Background - iterative feature transformation

Recall that IterFT is a feature transformation for style transfer that has been shown to support controlling the balance between content preservation and style transferal. It does so by making the final feature's second-order statistic resemble that of the style feature while maintaining its proximity to the content feature.

Formally, following Fig. 2(a), we let $\mathbf{F}_t$ be the feature transformed from the content feature $\mathbf{F}_c$ with reference to the style feature $\mathbf{F}_s$. For simplicity, a feature $\mathbf{F}$, which is a tensor of shape $C \times H \times W$ (C, H, W: channel length, height, width of $\mathbf{F}$) when produced from a neural network layer, is reshaped to a matrix of shape $C \times HW$. Iterative feature transformation (IterFT) [5] makes the second-order statistic, Gram matrix $\frac{1}{H_c W_c}\mathbf{F}_t\mathbf{F}_t^{\mathrm{T}}$, of $\mathbf{F}_t$ close to that of $\mathbf{F}_s$ while maintaining the proximity of $\mathbf{F}_t$ to the content feature $\mathbf{F}_c$. Letting $n_c = H_c W_c$ and $n_s = H_s W_s$, IterFT solves the optimization problem in Eq. 1 for $\mathbf{F}_t$ using gradient descent with the analytical gradient $\frac{\mathrm{d}l}{\mathrm{d}\mathbf{F}_t}$ in Eq. 2.

$$\min_{\mathbf{F}_t} l(\mathbf{F}_t) = \min_{\mathbf{F}_t} ||\mathbf{F}_t - \mathbf{F}_c||_2^2 + \lambda||\frac{1}{n_c}\mathbf{F}_t\mathbf{F}_t^{\mathrm{T}} - \frac{1}{n_s}\mathbf{F}_s\mathbf{F}_s^{\mathrm{T}}||_2^2, \quad (1)$$

$$\frac{\mathrm{d}l}{\mathrm{d}\mathbf{F}_t} = 2(\mathbf{F}_t - \mathbf{F}_c) + \frac{4\lambda}{n_c}(\frac{1}{n_c}\mathbf{F}_t\mathbf{F}_t^{\mathrm{T}} - \frac{1}{n_s}\mathbf{F}_s\mathbf{F}_s^{\mathrm{T}})\mathbf{F}_t, \quad (2)$$

where $\lambda > 0$ is the coefficient controlling the balance between content preservation and style transferal. With $\mathbf{F}_t$ initialized to $\mathbf{F}_c$, the final feature $\mathbf{F}_t$ is produced from n_{upd} iterations of the update rule: $\mathbf{F}_t \leftarrow \mathbf{F}_t - \eta\frac{\mathrm{d}l}{\mathrm{d}\mathbf{F}_t}$, where η is the learning rate.

Advanced photorealistic style transfer models often embed multiple feature transformations. This is exemplified in Fig. 2(b) for PhotoWCT^2 [6], which has four IterFTs. The first IterFT adapts the *relu4_1* content feature with respect to the *relu4_1* style feature for n_{upd} iterations, followed by the second IterFT which adapts the *relu3_1* content feature with respect to the *relu3_1* style feature for n_{upd} iterations until the last IterFT finishes adapting the *relu1_1* content feature with respect to the *relu1_1* style feature.

3.2. Modified iterative feature transformation

Our first modification is motivated by the observation that IterFT fails to stably produce reasonable results. This is exemplified in Fig. 3.

We hypothesize that IterFT's failures stem from the fact that it relies on only one step (i.e., second-order statistic

matching) rather than the three steps (i.e., centralization, second-order statistic matching, and decentralization) consistently employed by prior transformations [10, 15, 20] that stably produce reasonable results. Accordingly, we introduce centralization and decentralization before and after the second-order statistic matching. Our experimental results will validate the importance of these two steps by showing their addition to IterFT enables the resulting transformation to stably generate reasonable results (Sec. 4.2) and their removal from existing transformations (i.e., AdaIN [10] and ZCA [15]) lead to quality degradation in synthesized images (shown in the Supplementary Materials). We suspect that the theory motivating the benefit of this modification is closely related to mean vector matching. While prior works [8, 17] focus on explaining the reason of matching second-order statistics between style and stylized features for style transfer, we conjecture that matching first-order mean vectors is also important, and this is supported by centralization and decentralization. We explain how it is supported in the Supplementary Materials using the algorithm of the prior transformations [10, 15, 20].

Formally, let $\bar{\mathbf{F}} = \mathbf{F} - \mu(\mathbf{F})$ denote the centralized feature of $\mathbf{F}$, where $\mu(\mathbf{F})$ is the mean vector across the HW columns of $\mathbf{F}$ and the matrix-vector subtraction in $\mathbf{F}$ - $\mu(\mathbf{F})$ is done by array broadcasting. The algorithm applied by prior transformations [10, 15, 20] is as follows: (1) Centralization: centralize $\mathbf{F}_c$ and $\mathbf{F}_s$ to be $\bar{\mathbf{F}}_c$ and $\bar{\mathbf{F}}_s$. (2) Second-order statistic matching: alter $\bar{\mathbf{F}}_c$ to be $\bar{\mathbf{F}}_t$ such that for AdaIN [10] the variances of $\bar{\mathbf{F}}_s$ and $\bar{\mathbf{F}}_t$ are equal, and for ZCA [15] and OST [20] the covariances of $\bar{\mathbf{F}}_s$ and $\bar{\mathbf{F}}_t$ are equal. (3) Decentralization: add $\mu(\mathbf{F}_s)$ to $\bar{\mathbf{F}}_t$ to derive the transformed feature $\mathbf{F}_t$.

We modify IterFT to apply centralization before and decentralization after the iterative feature update. Mathematically, with the centralized content and style features $\bar{\mathbf{F}}_c$ and $\bar{\mathbf{F}}_s$, the modified optimization problem is written as Eq. 3, where the constraint $\mu(\bar{\mathbf{F}}_t) = \vec{0}$ requires $\bar{\mathbf{F}}_t$ to be centralized. With $\bar{\mathbf{F}}_t$ initialized as $\bar{\mathbf{F}}_c$, the centralized feature $\bar{\mathbf{F}}_t$ can be solved using gradient descent[3] in Eq. 4 with the analytical gradient $\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t}$ in Eq. 5.

$$\min_{\bar{\mathbf{F}}_t} l(\bar{\mathbf{F}}_t) = \min_{\bar{\mathbf{F}}_t} ||\bar{\mathbf{F}}_t - \bar{\mathbf{F}}_c||_2^2 + \lambda||\frac{1}{n_c}\bar{\mathbf{F}}_t\bar{\mathbf{F}}_t^{\mathrm{T}} - \frac{1}{n_s}\bar{\mathbf{F}}_s\bar{\mathbf{F}}_s^{\mathrm{T}}||_2^2 \quad \text{subject to } \mu(\bar{\mathbf{F}}_t) = 0, \tag{3}$$

$$\bar{\mathbf{F}}_t \leftarrow \bar{\mathbf{F}}_t - \eta\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t},\ \bar{\mathbf{F}}_t \leftarrow \bar{\mathbf{F}}_t - \mu(\bar{\mathbf{F}}_t), \tag{4}$$

$$\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t} = 2(\bar{\mathbf{F}}_t - \bar{\mathbf{F}}_c) + \frac{4\lambda}{n_c}(\frac{1}{n_c}\bar{\mathbf{F}}_t\bar{\mathbf{F}}_t^{\mathrm{T}} - \frac{1}{n_s}\bar{\mathbf{F}}_s\bar{\mathbf{F}}_s^{\mathrm{T}})\bar{\mathbf{F}}_t. \tag{5}$$

[3]While Quasi-Newton methods seem plausible, they are impractical here since they are memory-intensive due to computing Hessian matrices; e.g., for FHD input (1920x1080), the Hessian matrix of the loss function in Eq. 3 has (64x1920x1080)^2 = 1.76e^{16} elements at the *relu*1_1 layer.

Figure 3: Qualitative results exemplify our modification to IterFT overcomes the instability of IterFT in producing reasonable stylized images. These results are produced by embedding the transformations in the PhotoWCT^2 model.

The final resulting feature $\mathbf{F}_t$ is $\bar{\mathbf{F}}_t$ decentralized by $\mu(\mathbf{F}_s)$: $\bar{\mathbf{F}}_t + \mu(\mathbf{F}_s)$. We exemplify the improved quality resulting from our modification compared to IterFT in Fig. 3.

Notice that the constraint that $\mu(\bar{\mathbf{F}}_t) = \vec{0}$ is always satisfied while gradient descent minimizes the loss value. This can be shown by first assuming the current $\bar{\mathbf{F}}_t$ is centralized. With this, we have $\mu(\bar{\mathbf{F}}_t - \eta\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t}) = \mu(\bar{\mathbf{F}}_t) - \eta\mu(\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t})$ $= -\eta\mu(\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t})$. This implies the updated feature is centralized if $\mu(\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t}) = \vec{0}$, which can be shown as follows: $\mu(\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t})$ $= 2(\mu(\bar{\mathbf{F}}_t) - \mu(\bar{\mathbf{F}}_c)) + \mu(\mathbf{S}\bar{\mathbf{F}}_t) = \mu(\mathbf{S}\bar{\mathbf{F}}_t) = \mathbf{S}\mu(\bar{\mathbf{F}}_t) = \vec{0}$. Therefore, with $\bar{\mathbf{F}}_t$ initialized as $\bar{\mathbf{F}}_c$, which is centralized, the sequels of the updated $\bar{\mathbf{F}}_t$'s are all centralized.

3.3. Line search-based feature transformation

An issue that remains in our Modified IterFT is that the algorithm requires multiple iterations to update the feature, which is slow. In practice, the values of the learning rate η and the number of iterations n_{upd} are empirically set to be 0.01 and 15, respectively, in [5]. However, intuitively, the learning rate η should be dynamically determined such that η is larger in the beginning iterations to accelerate the convergence and smaller in the later to fine-tune the solution.[4] With a dynamic η, the number of iterations n_{upd} can then be greatly reduced. To dynamically determine the value of η for a new iteration with the latest $\bar{\mathbf{F}}_t$ from the last iteration and the derivative $\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t}$ calculated from Eq. 5, we solve the following line search optimization problem:

$$\min_{\eta} l(\bar{\mathbf{F}}_t - \eta\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t}) \text{ subject to } \eta > 0, \tag{6}$$

where the loss function l is defined in Eq. 3, and the constraint $\eta > 0$ forces $\bar{\mathbf{F}}_t$ to move toward the descent direction. The meaning of Eq. 6 is that we start from the point $\bar{\mathbf{F}}_t$ and search in the opposite direction of $\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t}$, i.e. the descent direction, to find a new point $\bar{\mathbf{F}}_t - \eta\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t}$ which minimizes

[4]While a trivial solution to accelerate convergence for existing methods could be to simply increase the learning rate, this is insufficient since we can't know how much to increase the rate. We discuss this in Supplementary Materials.

the loss function. With the substitution of $\frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t}$ in Eq. 6 with Eq. 5 and some arithmetic with calculus (detailed derivation is provided in Supplementary Materials), we can show that the optimal η should be a solution to this cubic equation:

$$a\eta^3 + b\eta^2 + c\eta + d = 0, \tag{7}$$

with the coefficients defined as follows:

$$a = \frac{2\lambda}{n_c^2}\mathrm{tr}[\mathbf{D_2}\mathbf{D_2}],\ b = -\frac{6\lambda}{n_c^2}\mathrm{tr}[\mathbf{D_F}\mathbf{D_2}],\ d = -\frac{1}{2}\mathrm{tr}[\mathbf{D_2}], \tag{8}$$

$$c = \mathrm{tr}[\mathbf{D_2}] + \frac{2\lambda}{n_c}\mathrm{tr}[\mathbf{D_2}\mathbf{S}] + \frac{2\lambda}{n_c^2}\big(\mathrm{tr}[\mathbf{D_F}\mathbf{D_F}] + \mathrm{tr}[\mathbf{D_F}\mathbf{D_F^{\mathrm{T}}}]\big), \tag{9}$$

where $\mathbf{D_2} \equiv \mathbf{D}\mathbf{D}^{\mathrm{T}}$, $\mathbf{D_F} \equiv \mathbf{D}\bar{\mathbf{F}}_t^{\mathrm{T}}$, $\mathbf{D} \equiv \frac{\mathrm{d}l}{\mathrm{d}\bar{\mathbf{F}}_t}$ and $\mathbf{S} \equiv \frac{1}{n_c}\bar{\mathbf{F}}_t\bar{\mathbf{F}}_t^{\mathrm{T}} - \frac{1}{n_s}\bar{\mathbf{F}}_s\bar{\mathbf{F}}_s^{\mathrm{T}}$.

Although Eq. 8 and Eq. 9 may look intimidating at the first glance, Eq. 7 can actually be solved efficiently for the following reasons. First, $\mathbf{D}$ and its sub-term $\mathbf{S}$ already must be computed for the gradient descent calculation in Modified IterFT, and so do not introduce extra overhead when line-searching η. Second, since all coefficients are rooted in only two repeated terms $\mathbf{D_2}$ and $\mathbf{D_F}$, we need to compute $\mathbf{D_2}$ and $\mathbf{D_F}$ just once to derive all coefficients. Third, the matrix multiplications $\mathbf{D_2}$ and $\mathbf{D_F}$ and the trace operation $\mathrm{tr}[\mathbf{AB}]$ of two matrices $\mathbf{A}$ and $\mathbf{B}$ can be computed in parallel with a GPU. Finally, with the coefficients computed, we can solve the cubic function in Eq. 7 in constant time using the cubic formula [29][5].

To comply with the constraint $\eta > 0$, we have to ensure that there is at least one positive solution to Eq. 7. We prove this in Supplementary Materials.

In summary, our line search-based feature transformation (LS-FT) produces the transformed feature $\mathbf{F}_t$ with the content and style features $\mathbf{F}_c$ and $\mathbf{F}_s$ in four steps:

1. Centralize the content and style features to be $\bar{\mathbf{F}}_c$ and $\bar{\mathbf{F}}_s$ and initialize $\bar{\mathbf{F}}_t$ to be $\bar{\mathbf{F}}_c$.
2. Calculate the gradient from Eq. 5 and the learning rate η from Eq. 7.
3. Update $\bar{\mathbf{F}}_t$ according to Eq. 4 and iterate from the step 2 if needed.
4. Decentralize $\bar{\mathbf{F}}_t$ by adding the mean $\mu(\mathbf{F}_s)$ of the style feature.

We will show in Sec. 4.1 that, unlike Modified IterFT, iteration in step 3 is not necessary and one feature update is sufficient for LS-FT. To balance the magnitude of the content loss $||\bar{\mathbf{F}}_t - \bar{\mathbf{F}}_c||_2^2$ and style loss $\lambda||\frac{1}{n_c}\bar{\mathbf{F}}_t\bar{\mathbf{F}}_t^{\mathrm{T}} - \frac{1}{n_s}\bar{\mathbf{F}}_s\bar{\mathbf{F}}_s^{\mathrm{T}}||_2^2$ in Eq. 3, it is best to have the value of λ close to the ratio $\frac{||\bar{\mathbf{F}}_t - \bar{\mathbf{F}}_c||_2^2}{||\frac{1}{n_c}\bar{\mathbf{F}}_t\bar{\mathbf{F}}_t^{\mathrm{T}} - \frac{1}{n_s}\bar{\mathbf{F}}_s\bar{\mathbf{F}}_s^{\mathrm{T}}||_2^2}$. However, since $\bar{\mathbf{F}}_t$ is unknown, we replace it with $\mathbf{0}$ and set λ to be $\frac{||\bar{\mathbf{F}}_c||_2^2}{||\frac{1}{n_s}\bar{\mathbf{F}}_s\bar{\mathbf{F}}_s^{\mathrm{T}}||_2^2}$ for content-style balance.

For tuning the content style balance (i.e., determined by the value of λ), we introduce a coefficient α such that $\lambda = \alpha\frac{||\bar{\mathbf{F}}_c||_2^2}{||\frac{1}{n_s}\bar{\mathbf{F}}_s\bar{\mathbf{F}}_s^{\mathrm{T}}||_2^2}$. Varying α in turn supports boosting the stylization strength or content preservation. We show results in the main paper for fixed α's and expanded analysis of the effect of α in the Supplementary Materials.

[5]Eq. 6 is a quartic function of η. If there are three positive solutions to Eq. 7, they correspond to a local minimum, a local maximum, and the global minimum of Eq. 6. Therefore, we just plug the solutions to Eq. 6 to see which one results in the lowest value and that is the one we pick.

4. Experiments

We now evaluate our transformation's ability to generalize in establishing a style-content balance across multiple photorealistic style transfer architectures (Sec. 4.2) and its improved computational efficiency (Sec. 4.3).

4.1. Convergence of line search optimization

We first describe our analysis to establish how many iterations to use for our line search optimization in our LS-FT transformation, by embedding it in four style transfer architectures: WCT^2, PhotoWCT, PhotoWCT^2, and PCA-d.

Implementations. The baseline is Modified IterFT, which we accelerate by introducing line search optimization to it. Modified IterFT follows IterFT [5] to adopt 0.01 as the learning rate and 15 as the number of iterations for each layer. We want to know how many iterations are required for LS-FT to outperform the performance of Modified IterFT after 15 iterations.

Dataset. We use the PST dataset [30], which is the largest dataset for photorealistic style transfer evaluation. It consists of 786 pairs of a content image and a style image.

Metric. Recall that both Modified IterFT and LS-FT iteratively adapt the content feature with respect to the style feature at each *reluN_1* layer for 15 iterations to minimize the loss value defined in Eq. 3. We monitor the loss value after each iteration of a transformation. Specifically, for each transformation there is a series of 15 loss values calculated for each *reluN_1* layer. Taking all input pairs into account, we have 786 series of 15 loss values at each *reluN_1* layer. We compute the mean and standard deviation across the 786 series and plot the mean series as a curve and the standard deviation series as a shaded area around the curve.

Results. Mean loss curves and associated standard deviations are shown in Fig. 4 for each of the four *reluN_1* layers for PhotoWCT^2. Due to limited space and similar findings, results for WCT^2, PhotoWCT, and PCA-d are provided in the Supplementary Materials.

When comparing LS-FT to Modified IterFT, we observe LS-FT converges much faster. For example, observing the mean curves at the *relu4_1* layer, LS-FT converges in only

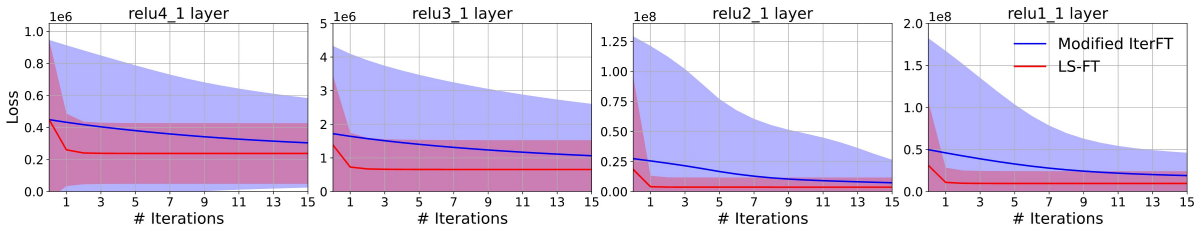

Figure 4: Convergence comparison between Modified IterFT and LS-FT. Here Modified IterFT and LS-FT are tested on the PhotoWCT^2 model [6] which applies feature transformations at the bottleneck (*relu4_1* layer) and the *relu3_1*, *relu2_1*, *relu1_1* layers in the decoder. For a content-style input pair, a loss value is calculated according to Eq. 3 after each iteration of Modified IterFT or LS-FT at each layer, resulting in a series of 15 loss values for 15 iterations. A curve shows the mean series across 786 series from all 786 input pairs. The surrounding shaded area indicates the region within one standard deviation. It is observed that LS-FT converges faster than Modified IterFT and only one iteration is sufficient for LS-FT to outperform Modified IterFT at each layer.

two iterations with the second iteration slightly improving from the first while Modified IterFT slowly converges over all iterations. Moreover, the loss value of LS-FT after the first iteration is already lower than that of Modified IterFT after 15 iterations, suggesting that one iteration is sufficient for LS-FT at the *relu4_1* layer. Note that the smaller standard deviation of LS-FT than that of Modified IterFT at each iteration implies the faster convergence of LS-FT is universal across the dataset. We notice the same phenomenon occurs at the other three layers, suggesting one feature update is enough for LS-FT at these layers as well.

4.2. Content preservation and stylization strength

To evaluate our feature transformation's ability to achieve a better balance between content preservation and stylization strength than prior transformations, we benchmark multiple transformations embedded in four autoencoder-based style transfer models: WCT^2 [31], PhotoWCT [16], PhotoWCT^2 [6], and PCA-d [7].

Dataset. We again test with the PST dataset [30], which consists of 786 pairs of a content image and a style image.

Our Implementations. We again evaluate our LS-FT and Modified IterFT (i.e., our stable IterFT [5]).

Baselines. For comparison, we evaluate ZCA [15], OST [20], AdaIN [10], and MAST [11]. IterFT [5] is not used for comparison since it results in many unreasonable results, as is shown in the Supplementary Materials.

Metrics. For each model-transformation pair, we compute the mean content loss and mean style loss across all stylized images by computing for each stylized image I_{sty}, its content loss from the content image I_c and style loss from the style image I_s. To define the losses, we let $\bar{\mathbf{F}}_{k,N} \in C_N \times H_{k,N}W_{k,N}$ $(N = 1, 2, 3, 4)$ denote the centralized *reluN_1* feature of the image I_k $(k \in \{c, s, sty\})$. Following NST [8], which uses the *relu4_1* layer to represent content and multiple layers (e.g. *reluN_1*, N $= 1, 2, 3, 4$) to represent style, we define the content loss as $||\bar{\mathbf{F}}_{sty,4} - \bar{\mathbf{F}}_{c,4}||_2^2$, and the style loss as $\sum_{N=1}^{4} ||\frac{1}{H_{c,N}W_{c,N}}\bar{\mathbf{F}}_{sty,N}\bar{\mathbf{F}}_{sty,N}^{\mathrm{T}} - \frac{1}{H_{s,N}W_{s,N}}\bar{\mathbf{F}}_{s,N}\bar{\mathbf{F}}_{s,N}^{\mathrm{T}}||_2^2$.

Then, we evaluate the quality of stylized images using image quality assessment metrics: SSIM [28], FSIM [32], and NIMA [27]. SSIM and FSIM assess the structural similarity between a stylized image and its source content image, while NIMA evaluates the stylized image as a standalone photo without referencing to the content image.

Results. Fig. 5(a) shows the overall distribution of mean content-style losses resulting from different model-transformation pairs. We observe that MAST results in the weakest style effects across all models, which is reinforced by qualitative results shown in Fig. 6. Consequently, for our fine-grained analysis with respect to each model (Fig. 5(b,c,d)), we exclude MAST from the analysis.

As shown in Fig. 5(b,c,d), our Modified IterFT and LS-FT have similar performance and consistently lead to a better balance between content preservation and stylization strength than prior transformations for all four style transfer architectures. For WCT^2, its low content losses in Fig. 5(a) come from its model design while it relies on a transformation to boost its stylization strength to faithfully reflect the style. We observe that LS-FT with $\alpha = 10$ successfully boosts the stylization strength of WCT^2 compared to other transformations (Fig. 5(b)), with AdaIN resulting in a 30.7% greater style loss, and ZCA resulting in a 6.8% greater style loss. Their worse style losses are also reflected in the qualitative results. As exemplified in the first row of Fig. 6, while AdaIN poorly transfers the red leaf effect and ZCA transfers it more strongly, LS-FT improves upon ZCA by rendering more red color to the leaves.

The PhotoWCT model is designed to reflect the style well and so have lower style losses, as demonstrated in Fig. 5(a). Consequently, it needs a transformation to boost its content preservation. Quantitatively, our transformations (e.g., LS-FT with $\alpha = 0.2$) result in the lowest content loss compared to other transformations (Fig. 5(c)), with ZCA resulting in a 62.1% greater content loss, and AdaIN result-

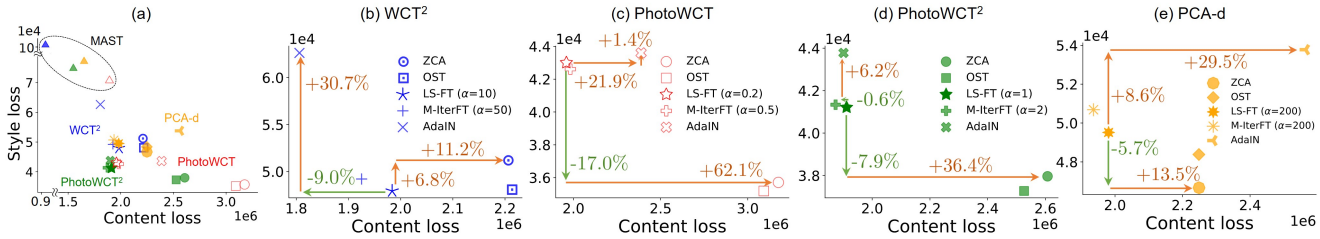

Figure 5: Mean content losses vs. mean style losses of stylized images for different model-transformation pairs. Unlike other transformations, our Modified IterFT (M-IterFT) and LS-FT boosts WCT^2's stylization strength, enhances PhotoWCT and PCA-d's content preservation, and reaches a good balance between style strength and content preservation for PhotoWCT^2.

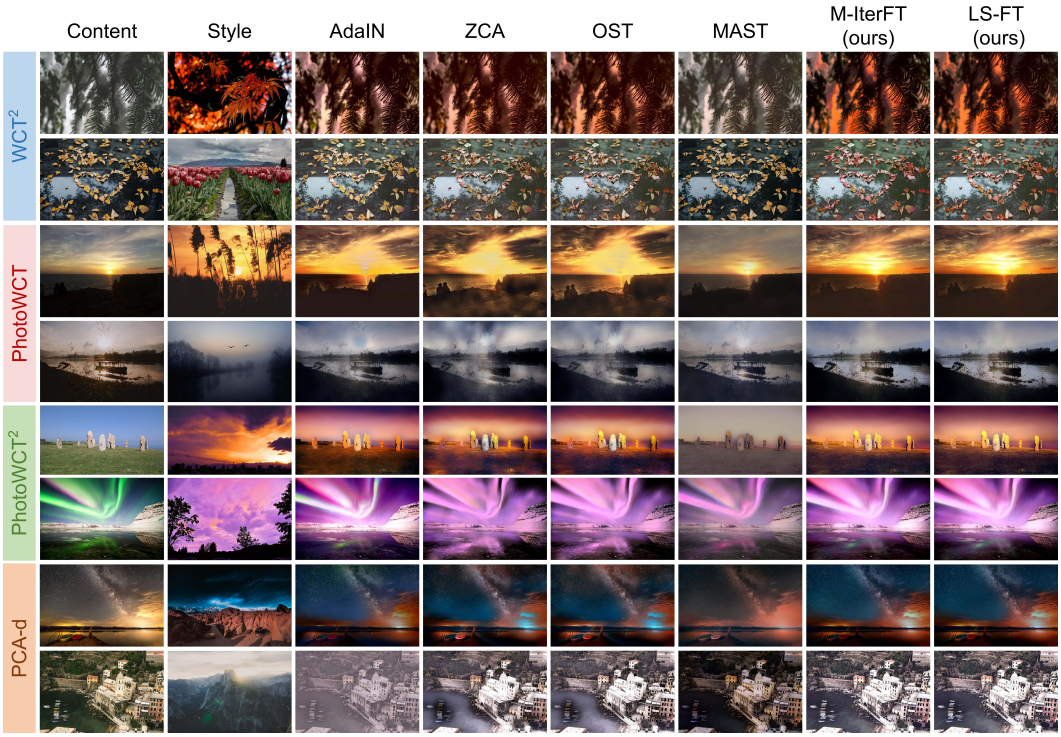

Figure 6: Examples of a better content-style control from our transformations: Modified IterFT (M-IterFT) and LS-FT. Unlike other transformations, Modified IterFT and LS-FT can generalize to different models to achieve content-style balance.

ing in a 21.9% greater content loss. Qualitatively, if we take the first row of the PhotoWCT panel in Fig. 6 for example, 62.1% greater content loss of ZCA results in severe artifacts (uneven reflection of sunlight on the ground), while 21.9% greater content loss of AdaIN makes it fail to preserve the finer content (less realistic sunset in the background).

The PhotoWCT^2 model is designed to improve the content preservation from PhotoWCT and the stylization strength from WCT^2. However, it does not always preserve content well and transfer enough style effects, as mentioned before in Fig. 1 where it transfers insufficient style effects with AdaIN and introduces artifacts that ruin the content with ZCA. Thus, it needs a transformation to result in stronger stylization strength than AdaIN and better content preservation than ZCA. We observe our LS-FT with $\alpha = 1$ can achieve this requirement by reducing the style loss of AdaIN by 6.2% and the content loss of ZCA by 36.4% (Fig. 5(d)). As exemplified in the first row of the PhotoWCT^2 panel in Fig. 6, our LF-FT preserves better content than ZCA and transfers stronger style than AdaIN.

PCA-d is an improved version of PhotoWCT^2 in that it is lightweight and achieves better content-style balance than PhotoWCT^2. While [7] shows that PCA-d produces realistic results and reflects good style effects when ZCA is used as the transformation, our experiment shows that ZCA can still produce slight artifacts (unnatural sunlight in the first row of the PCA-d panel in Fig. 6) and severe artifacts occasionally (misty artifacts on the lake in the second row). As

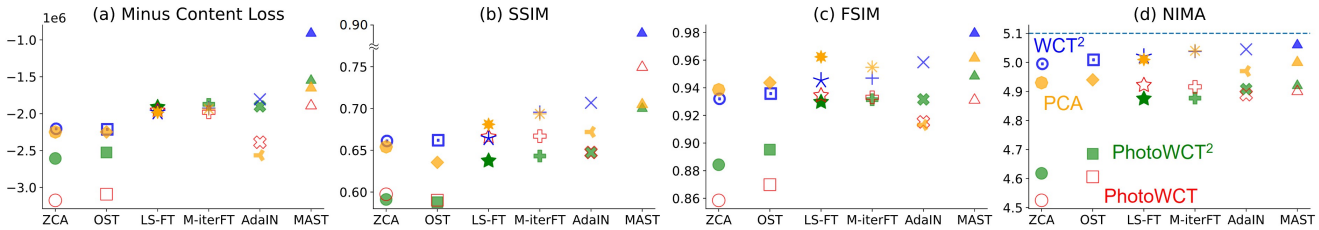

Figure 7: Quality assessment of stylized images resulting from different model-transformation pairs. For each model, compared to the popular AdaIN and ZCA, our Modified IterFT and LS-FT can result in a score that is comparable to that of AdaIN and higher than that of ZCA. Note that the dash line in (d) indicates the average NIMA score for content imges in the PST dataset.

shown in Fig. 5(e), AdaIN performs even worse than ZCA with worse content and style losses. Our LS-FT, in contrast, mitigates the artifacts by reducing the content loss of ZCA by 13.5%, as exemplified in Fig. 6.

The mean quality scores of stylized images resulting from different model-transformation pairs reinforce our aforementioned findings (Fig. 7). In particular, for PhotoWCT and PCA-d the quality scores of LS-FT and Modified IterFT are the highest of all transformations except MAST due to the boost of content preservation, while in the cases of the other two models, LS-FT, Modified IterFT, and AdaIN have comparable quality scores which are higher than those from ZCA and OST. Note that MAST tends to have higher scores since it usually weakly adapts content.

4.3. Speed performance

We finally compare the speed of our LS-FT to other transformations, when embedded in four style transfer architectures: WCT^2 [31], PhotoWCT [16], PhotoWCT^2 [6], and PCA-d [7]. All tests are conducted with an Nvidia-1080Ti GPU with 11GB memory.

Our Implementations. We evaluate our LS-FT transformation and its ablated variant that lacks line search for the speed up: Modified IterFT (stable IterFT [5]).

Baselines. We evaluate ZCA [15], OST [20], AdaIN [10], and MAST [11].[6]

Dataset. We test on four resolutions: 1280×720 (HD), 1920×1080 (FHD), 2560×1440 (QHD), and 3840×2160 (UHD or 4K). To collect images, we downloaded a 4K video [25] from YouTube, sampled 100 frames, and downsampled each frame to the other lower resolutions. For each model, the speed of a transformation for each resolution is averaged across the total 100 images.

Results. Tab. 2 shows the stylization speeds. We report results from PCA-d in another table in the Supplementary Materials because PCA-d is a distilled model that produces lightweight features leading to faster transformations and because of limited space.

[6]IterFT has almost the same speed of M-IterFT and so is ignored here.

	WCT^2 / PhotoWCT / PhotoWCT^2	Not Tunable				Tunable	
		ZCA	OST	AdaIN	MAST	M-IterFT	LS-FT
HD	0.18 / 0.37 / 0.13	0.18	0.27	**0.01**	0.60	0.29	**0.04**
FHD	0.59 / 0.63 / 0.24	0.20	0.29	**0.02**	1.14	0.73	**0.09**
QHD	OOM / 0.98 / 0.40	0.24	0.33	**0.06**	2.15	1.27	**0.17**
UHD	OOM / OOM / 0.88	0.33	0.40	**0.13**	5.02	2.84	**0.34**

Table 2: The speeds for stylization of images of different resolutions using different transformations. For clarity, the time spent on the model and the transformation is separated. The transformations are categorized into two groups based on whether they consider model adpativeness or not. LS-FT is consistently 7-8x times faster than Modified IterFT (M-IterFT) in all resolutions due to no need of multiple iterations. LS-FT also runs faster than or comparably to ZCA and OST. OOM: Out of Memory. Unit: Second.

Compared to its ablated variant Modified IterFT which also controls the balance between stylization strength and content preservation, LS-FT is consistently 7-8x times faster due to no need of multiple iterations. For example, it takes LS-FT 0.34 seconds to stylize a UHD image, which is 8.35x faster than the 2.84 seconds for Modified IterFT.

When comparing LS-FT to the four baseline transformations lacking control over the balance between stylization strength and content preservation, overall LS-FT is competitive. For example, LS-FT is faster than OST and MAST in all resolutions, while LS-FT is comparably fast to ZCA in the UHD resolution case and faster than ZCA in the others. The one exception is AdaIN, which is the fastest transformation. This is due to its simplest math formulation.

5. Conclusion

We derived a new line search-based feature transformation (LS-FT) for photorealistic style transfer. Experiments show LS-FT with different style transfer architectures outperforms existing transformations by either boosting stylization strength, enhancing photorealism, or reaching a better style-content balance, while running at a fast speed.

References

[1] Jie An, Haoyi Xiong, Jun Huan, and Jiebo Luo. Ultrafast photorealistic style transfer via neural architecture search. In *AAAI*, pages 10443–10450, 2020.

[2] Ivan Anokhin, Pavel Solovev, Denis Korzhenkov, Alexey Kharlamov, Taras Khakhulin, Aleksei Silvestrov, Sergey Nikolenko, Victor Lempitsky, and Gleb Sterkin. High-resolution daytime translation without domain labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7488–7497, 2020.

[3] Jiaxin Cheng, Ayush Jaiswal, Yue Wu, Pradeep Natarajan, and Prem Natarajan. Style-aware normalized loss for improving arbitrary style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 134–143, 2021.

[4] Tai-Yin Chiu. Understanding generalized whitening and coloring transform for universal style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4452–4460, 2019.

[5] Tai-Yin Chiu and Danna Gurari. Iterative feature transformation for fast and versatile universal style transfer. In *European Conference on Computer Vision*, pages 169–184. Springer, 2020.

[6] Tai-Yin Chiu and Danna Gurari. Photowct2: Compact autoencoder for photorealistic style transfer resulting from blockwise training and skip connections of high-frequency residuals. *arXiv preprint arXiv:2110.11995*, 2021.

[7] Tai-Yin Chiu and Danna Gurari. Pca-based knowledge distillation towards lightweight and content-style balanced photorealistic style transfer models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7844–7853, 2022.

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

[9] Kibeom Hong, Seogkyu Jeon, Huan Yang, Jianlong Fu, and Hyeran Byun. Domain-aware universal style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14609–14617, 2021.

[10] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[11] Jing Huo, Shiyin Jin, Wenbin Li, Jing Wu, Yu-Kun Lai, Yinghuan Shi, and Yang Gao. Manifold alignment for semantically aligned style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14861–14869, 2021.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[13] Sunwoo Kim, Soohyun Kim, and Seungryong Kim. Deep translation prior: Test-time training for photorealistic style transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1183–1191, 2022.

[14] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3809–3817, 2019.

[15] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in neural information processing systems*, pages 386–396, 2017.

[16] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 453–468, 2018.

[17] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2230–2236, 2017.

[18] Jie Liang, Hui Zeng, and Lei Zhang. High-resolution photorealistic image translation in real-time: A laplacian pyramid translation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9392–9400, 2021.

[19] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, 30, 2017.

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

[21] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4990–4998, 2017.

[22] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018.

[23] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European conference on computer vision*, pages 319–345. Springer, 2020.

[24] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33:7198–7211, 2020.

[25] PK ski. See the most beautiful villages in switzerland: lakes, mountains, and green rolling hills. `https://www.youtube.com/watch?v=Ww4Wc34s-fA`. Accessed: 2021-10-15.

[26] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021.

[27] Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE Transactions on Image Processing*, 27(8):3998–4011, 2018.

[28] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[29] Wikipedia. Cubic equation — general cubic formula. `https://en.wikipedia.org/wiki/Cubic_equation#General_cubic_formula`. Accessed: 2021-09-25.

[30] 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. *arXiv preprint arXiv:2004.10955*, 2020.

[31] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9036–9045, 2019.

[32] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011.