

# AdaCM: Adaptive ColorMLP for Real-Time Universal Photo-Realistic Style Transfer

Tianwei Lin<sup>1</sup>\*, Honglin Lin<sup>2\*†</sup>, Fu Li<sup>1</sup>, Dongliang He<sup>1</sup>, Wenhao Wu<sup>3,1</sup>, Meiling Wang<sup>1</sup>,  
Xin Li<sup>1</sup>, Yong Liu<sup>2</sup>

<sup>1</sup>Baidu Inc.,

<sup>2</sup>Zhejiang University,

<sup>3</sup>The University of Sydney

wzmsltw@gmail.com, linhl97@zju.edu.cn

## Abstract

Photo-realistic style transfer aims at migrating the artistic style from an exemplar style image to a content image, producing a result image without spatial distortions or unrealistic artifacts. Impressive results have been achieved by recent deep models. However, deep neural network based methods are too expensive to run in real-time. Meanwhile, bilateral grid based methods are much faster but still contain artifacts like overexposure. In this work, we propose the **Adaptive ColorMLP (AdaCM)**, an effective and efficient framework for universal photo-realistic style transfer. First, we find the complex non-linear color mapping between input and target domain can be efficiently modeled by a small multi-layer perceptron (ColorMLP) model. Then, in **AdaCM**, we adopt a CNN encoder to adaptively predict all parameters for the ColorMLP conditioned on each input content and style image pair. Experimental results demonstrate that AdaCM can generate vivid and high-quality stylization results. Meanwhile, our AdaCM is ultrafast and can process a 4K resolution image in 6ms on one V100 GPU.

## Introduction

Photo-realistic style transfer is an attractive technique that migrates the artistic style of an example style image to a content image, generating an image as plausible as taken by a real camera. However, when transferring color, existing artistic style transfer methods (Gatys, Ecker, and Bethge 2016; Johnson, Alahi, and Fei-Fei 2016; Park and Lee 2019) will also transfer local texture patterns from style image to content image, resulting in severe spatial distortions and unrealistic artifacts, which are not favored. To address the spatial distortion problem in photo-realistic stylization, Luan et al. (Luan et al. 2017) introduce a photorealism constraint on optimization-based stylization method (Gatys, Ecker, and Bethge 2016). However, its expensive computation costs largely limit its practical applications. To overcome this issue, Li et al. (Li et al. 2018b) and Li et al. (Li et al. 2018a) both adopt “feed-forward and post-processing” framework.

First, PhotoWCT (Li et al. 2018b) and Linear (Li et al. 2018a) are proposed with relatively weak spatial distortions. Then, post-processing steps are applied to reduce remaining artifacts (optimization-based smoothing used in (Li et al. 2018b) and spatial propagation network used in (Li et al. 2018a)). Yoo et al. (Yoo et al. 2019) then propose an end-to-end deep model WCT<sup>2</sup>, which keeps local structure via multi-scales skip connections of wavelet transform. However, these methods are still too slow and occupy too large GPU memory to process very high-resolution images in real-time.

Recently, Xia et al. (Xia et al. 2020) achieve real-time stylization on 4K resolution (12 Mega-pixels) images, where the mapping between pixel’s input and output color is modeled as a  $3 \times 4$  affine transform. The proposed method first predicts low-resolution affine coefficient matrices in the bilateral grid, then it uses a full-resolution single-channel guidance map to slice these matrices into full-resolution coefficient matrices that are applied to each pixel of the original image. However, the  $3 \times 4$  color affine transform process has limited capability especially for complex color mapping relationship, which may lead to overexposure in stylization results. Besides, Xia et al. (Xia et al. 2020) may cause the contrast reduction in some cases, especially along strong edges, which is a known limitation of the local affine transform model (Chen et al. 2016).

Unlike predicting simple affine transform locally for different location in image, we aim to model stylization with global and expressive color mapping models in this work. *What is a more capable model for color mapping?* A widely used and expressive tool is 3D look-up table (3D-LUT), which is very fast and can describe complex color mapping relationship accurately. However, as a large three-dimension matrix, 3D-LUTs generally need elaborate adjustments of experts, and are fixed after the adjustment. In essence, 3D-LUTs describe complex color mapping relationship in an exhaustive enumeration way with large parameter redundancy. In this work, we propose to use a **Color Multi-Layer Perceptron (ColorMLP)** to model non-linear color mapping relationship and find out a small ColorMLP can accurately approximate most of 3D-LUTs with only 0.5% of their parameters.

\*These authors contributed equally.

†This work was done when Honglin Lin was an intern at Baidu. Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



Figure 1: Illustration of our results on image/video photo-realistic style transfer. We highly recommend Adobe Acrobat to view the animated video clips at the right side.

This lightweight property leaves space for us to learn to predict parameters for ColorMLP, thus enabling adaptive color manipulation. In this work, we propose a simple, effective and efficient framework **Adaptive ColorMLP (AdaCM)** for photo-realistic stylization. First, an encoder is adopted to combine context of low-resolution content and style images and predict MLP parameters. Then, we use the ColorMLP with the predicted parameters to transfer color of each pixel in the original content image. Furthermore, during inference, we use AdaCM to generate a 3D-LUT as intermediary, then can process 4K resolution images in real-time. To summarize, the main contributions are as follows:

1. We propose to use a lightweight and expressive ColorMLP model to approximate a fixed 3D-LUT, where a large proportion of parameters can be saved. This property enables adaptive color manipulation.
2. We introduce a novel Adaptive ColorMLP framework for photo-realistic stylization, where per-pixel color mapping is modeled using a lightweight MLP with adaptively predicted parameters.
3. Experiments demonstrate that our method can generate high-quality photo-realistic stylization results without spatial distortions, meanwhile, it is efficient to process 4K resolutions images in real-time.

## Related Work

### Photo-realistic Style Transfer

Inspired by the seminal work of Gatys et al. (Gatys, Ecker, and Bethge 2016), significant advances have been made in artistic style transfer (Johnson, Alahi, and Fei-Fei 2016; Li and Wand 2016; Huang and Belongie 2017; Li et al. 2017; Sheng et al. 2018; Park and Lee 2019) in the past few years. While producing impressive results, these methods tend to generate severely distorted semantic structure, which is visually unpleasant in photo-realistic stylization. To achieve photo-realistic stylization, most early works align statistics of color distribution, including their means and variances in pixel space (Reinhard et al. 2001) and histograms (Pitie, Kokaram, and Dahyot 2005; Xiao and Ma 2009). However, these global transformations ignore regional characteristics, which only transfer simple styles like color and tone. Although some methods have been proposed to model dense mapping locally (Shih et al. 2013, 2014; Wu et al. 2013; Tsai et al. 2016), they are restricted to specific scenarios.

Based on Gatys’ neural algorithm, Luan et al. (Luan et al. 2017) added a regularization term to obtain local affine color transformation, but it needs heavy computation to remove non-photorealistic artifacts due to the iterative optimization process. Li et al. (Li et al. 2018b) and Li et al. (Li et al. 2018a) both adopted post-processing (optimization-based smoothing used in (Li et al. 2018b) and SPN (Liu et al. 2017) used in (Li et al. 2018a)) for high quality results. Along another line of introducing more details of high frequencies into the decoder, WCT<sup>2</sup> (Yoo et al. 2019) and PhotoNAS (An et al. 2020) employed wavelet transform and skip connections respectively to utilize the information lost during feature extraction. However, all these methods need to process full-resolution images with deep neural network, and the inference speed grows rapidly with increasing of resolution. Therefore, they are limited in practical applications. Recently, Xia et al. (Xia et al. 2020) designed joint bilateral learning method to acquire local affine transforms on each spatial grid of deep features. They achieved real-time stylization on 4K resolution (12 Mega-pixels) but sometimes led to blur regions due to improper coefficient matrices.

### Color Enhancement

3D-LUTs are ultrafast operators to model color mapping relationship, and are widely used to improve perceptual quality of digital images. Adaptive 3D-LUT (Zeng et al. 2020) learned 3D-LUTs from training dataset and dynamically predicted their fusion weights, which can better capture the joint distribution between channels compared with the curve-based transformation (Guo et al. 2020; Kim, Koh, and Kim 2020a,b; Song, Qian, and Du 2021) and is much more efficient than the local affine matrix (Gharbi et al. 2017). However, since the basic sets of 3D-LUTs are fixed after training, adaptive 3D-LUTs cannot achieve universal style transfer, when the color mapping relationships vary largely for different content and style images. Recently, Liu et al. (Liu et al. 2021) enhanced the color of each pixel in image with a conditioned MLP, where the condition vector is generated by a shallow encoder. However, the parameters of MLP are still fixed after training. Differently, we propose to directly predict the full parameters of ColorMLP, thus can model a large variety of color mapping relationships corresponding to different input image pairs.

## Method

In this section, we first give a brief description of 3D-LUT and ColorMLP, and discuss the fitting capability of ColorMLP. Then, we introduce the proposed universal photo-realistic stylization framework AdaCM in detail.

### 3D-LUT and ColorMLP

**3D-LUT.** 3D lookup table is an effective tool for color transfer, which can explicitly preserve complex color mapping relationship, and is widely used in many image retouching applications. The parameters of a 3D-LUT can be denoted as  $V \in R^{M \times M \times M \times 3}$ , where  $M$  can be regarded as the number of bins and it determines the precision of transformation. Empirically,  $M$  is always set to 33 in practice. As introduced

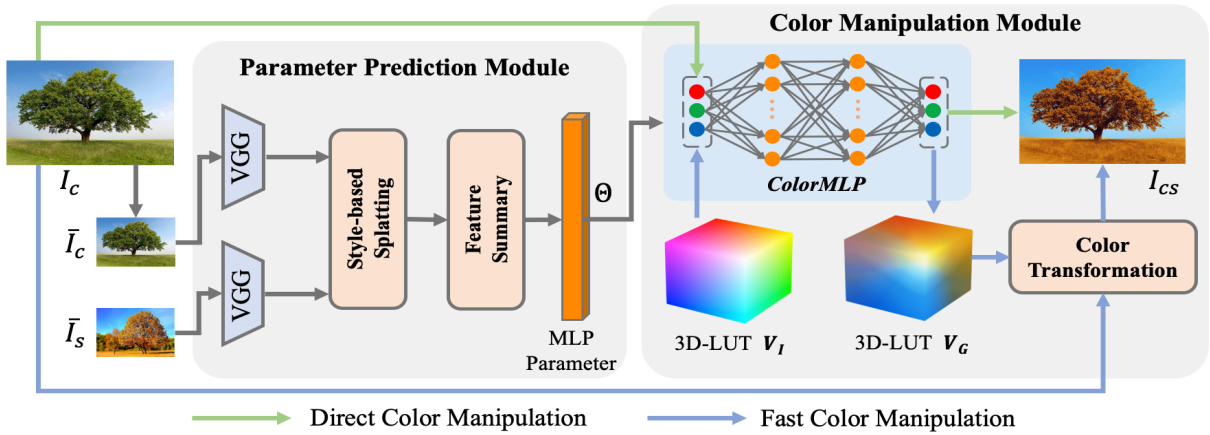


Figure 2: Overview of the full framework of AdaCM. (1) We first down-sample the content and style images to lower resolution, and feed them into the Parameter Prediction Module (PPM) to predict the parameter for ColorMLP. (2) We propose two color manipulation pipelines. The “Direct” pipeline is used during training where each pixel is directly processed by ColorMLP. The “Fast” pipeline is used during inference, where ColorMLP is used to generate a 3D-LUT, and then this 3D-LUT is used to transfer pixel color more efficiently.

in (Zeng et al. 2020), the color transformation in 3D-LUT is achieved by two efficient operations: lookup and trilinear interpolation. Take a pixel  $p = \{r, g, b\}$  as input, the color transformation process of 3D-LUT can be denoted as:

$$p' = LUT(p, V), \quad (1)$$

**ColorMLP.** In this work, we explore to utilize MLP to model complex color mapping with fewer parameters. As shown in Fig. 2, the proposed ColorMLP model has two hidden layers, each contains  $N$  units. Thus, the weight and bias parameters of a ColorMLP can be denoted as a vector  $\Theta \in R^D$ , where  $D = N^2 + 8N + 3$ . And the corresponding color transformation process is:

$$p' = ColorMLP(p, \Theta). \quad (2)$$

**Fitting Capability of ColorMLP.** Can we use ColorMLP to accurately approximate a given 3D-LUT? To achieve this, we first randomly sample a large amount of colors as input and obtain target colors using 3D-LUT. Then, we use these generated color pairs to train our ColorMLP model. After convergence, we uniformly sample  $M^3$  colors, and calculate the average L1 error between colors transferred by ColorMLP and 3D-LUT. We conduct experiments on 10 collected 3D-LUTs of different styles, and take average error as results. For each 3D-LUT, we train ColorMLP with different number of hidden units. The experimental results are shown in Fig. 3, which demonstrates that larger  $N$  can bring lower error. We set the pixel value range as 0~255 during evaluation. When  $N = 20$ , the pixel-wise L1 error is only 2.69, which is already hard for human to distinguish the differences. Thus, a two-layer ColorMLP with  $N = 20$  units is enough to fit a 3D-LUT. This lightweight property (0.5% parameters compared to 3D-LUT) leaves us space to further devise adaptive color transformation.

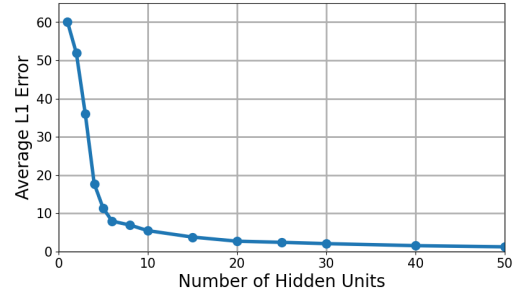


Figure 3: Average L1 error versus number of hidden units in MLP.

### Adaptive ColorMLP

Based on the lightweight property of ColorMLP, we propose the Adaptive ColorMLP framework to achieve universal photo-realistic style transfer. As shown in Fig. 2, our proposed AdaCM takes a content image  $I_c$  and an arbitrary style image  $I_s$  as inputs, and eventually synthesizes a stylized image  $I_{cs}$ . To achieve this, at first, we down-sample the input images to low resolution counterparts  $\bar{I}_c$  and  $\bar{I}_s$ . Then, a *Parameter Prediction Module (PPM)* will combine the feature statistics of  $\bar{I}_c$  and  $\bar{I}_s$ , and predicts the parameters  $\Theta$  for ColorMLP. This process can be formulated as:

$$\Theta = PPM(\bar{I}_c, \bar{I}_s), \quad (3)$$

In the final step, the color of original content image  $I_c$  is transferred based on the predicted parameter  $\Theta$  and generates stylized image  $I_{cs}$ .

### Parameter Prediction Module

The Parameter Prediction Module (PPM) aims at modeling the color mapping relationship between content and style images via predicting the parameters of ColorMLP. The core

is to match feature statistics of content image and style image. To achieve this, following (Xia et al. 2020), (1) we first adopt pre-trained VGG-19 (Simonyan and Zisserman 2015) network to extract multi-resolution feature maps of content and style images; (2) then we use multiple *Style-based Splatting* (Xia et al. 2020) blocks to fuse content and style feature maps based on AdaIN (Huang and Belongie 2017); (3) finally we adopt multiple convolution and fully-connection layers to summarize feature statistics and predict parameters  $\Theta$  of ColorMLP. In this way, we can fully fuse the content and style statistic in multiple feature granularity. Meanwhile, since PPM processes images in low-resolution, its computation cost is limited. More implementation details of PPM are provided in the supplementary material.

### Color Manipulation Module

As shown in Fig. 2, the direct way of color manipulation is to directly use ColorMLP with predicted parameters to process the color of each pixel. However, with the growth of resolution, there are too many pixels in an image. For example, a 4K resolution image contains around 12 Megapixels. Thus, although ColorMLP is extremely lightweight, the overall computation cost will still grow obviously with the increasing resolution. Meanwhile, 3D-LUT is ultrafast even under high resolution due to its  $O(1)$  complexity and 0 float-point operation in need for each input pixel. Thus, to further improve the inference efficiency, we propose a fast inference pipeline with the help of 3D-LUT. As shown in Fig. 2, we first use ColorMLP to generate an equivalent 3D-LUT as:

$$V_G = \text{ColorMLP}(V_I, \Theta), \quad (4)$$

where  $V_I$  is the parameters of identity 3D-LUT (each color maps to itself), and  $V_G$  is the parameters of generated 3D-LUT. Then, we conduct color transformation as:

$$I_{cs} = \text{LUT}(I_c, V_G). \quad (5)$$

As shown in Table. 3, with the growth of image resolution, this 3D-LUT based pipeline is extremely faster than directly transferring image color using ColorMLP. In addition, via generating  $V_G$ , we can also intuitively visualize the color mapping relationship modeled by ColorMLP.

### Loss Function

The proposed AdaCM is trained end-to-end since it is fully differentiable. Our overall loss function is the weighted summation of content loss  $\mathcal{L}_c$ , style loss  $\mathcal{L}_s$  and the regularization term  $R$ :

$$\mathcal{L}_{all} = \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s + \lambda_r \mathcal{R}_{all}, \quad (6)$$

where  $\lambda_c$ ,  $\lambda_s$  and  $\lambda_r$  are hyper-parameters controlling weights of their corresponding loss terms. Details of each term will be explained in the remaining part of this section.

**Content and Style Losses.** For content loss, following previous works (Huang and Belongie 2017; Park and Lee 2019), we adopt the commonly used normalized perceptual loss. The content loss is defined as:

$$\mathcal{L}_c = \sum_{i=1}^4 \left\| \text{norm}(F_c^i) - \text{norm}(F_{cs}^i) \right\|_2, \quad (7)$$

where  $F^i$  denotes the  $i$ -th layer of feature map extracted by pre-trained VGG model, and  $\text{norm}$  denotes the channel-wise normalization. For style loss, we adopt the commonly used mean-variance loss (Huang and Belongie 2017) as:

$$\mathcal{L}_s = \sum_{i=1}^3 \left( \left\| \mu(F_s^i) - \mu(F_{cs}^i) \right\|_2 + \left\| \sigma(F_s^i) - \sigma(F_{cs}^i) \right\|_2 \right), \quad (8)$$

where  $\mu$  and  $\sigma$  calculate the mean and variance of the feature vectors separately.

**Regularization.** Using content and style losses, we can train the AdaCM to achieve universal photo-realistic stylization. However, as shown in Fig. 6, generated results may contain banding artifacts or abrupt color changes in original smooth areas. indicating that smoothness is not carefully considered during stylization. To ensure the generated ColorMLP containing smooth color mapping, we first consider the total variation loss  $\mathcal{R}_{lut}$  on 3D-LUT as proposed in (Zeng et al. 2020), which penalizes differences between adjacent cells of the generated 3D-LUT. To achieve this, we first generate a 3D-LUT  $V_G$  using ColorMLP, and conduct the regularization  $\mathcal{R}_{lut}$  on 3D-LUT  $V_G$ . To be specific, we denote  $v = (i, j, k)$  as one point in  $V_G$  and its color is  $V_G(v)$ . Then,  $\mathcal{R}_{lut}$  can be defined as:

$$\mathcal{R}_{lut} = \sum_{v \in V_G} \sum_{v_n \in \Omega} \|V_G(v) - V_G(v_n)\|_2, \quad (9)$$

where  $\Omega = \{(i+1, j, k), (i, j+1, k), (i, j, k+1)\}$  denotes the neighbors of  $v$ . As shown in Fig. 6, artifacts are largely reduced but still exist in some scenarios. The main reason is that  $\mathcal{R}_{lut}$  smooths the local mapping relationship around all colors in color space, while the color distribution of a usual content image lies only a small part of color space. Thus, we further adopt a regular total variance loss  $\mathcal{R}_{img}$  on image  $I_{cs}$ , which focuses on the local mapping smoothness around colors appeared in content image rather than all colors. The overall regularization term is defined as:

$$\mathcal{R}_{all} = \mathcal{R}_{lut} + \mathcal{R}_{img} \quad (10)$$

## Experiment

### Dataset and Setup

**Dataset.** We combine all images from Landscape (Afifi, Brubaker, and Brown 2021) dataset and 6000 images randomly sampled from MS-COCO (Lin et al. 2014) dataset as our training set. During training, we randomly sample both content images and style images from this mixed dataset. During evaluation, we use high-quality test images collected by (Xia et al. 2020). Some copyright free images from Pexels<sup>1</sup> are also used during testing.

<sup>1</sup>www.pexels.com

Method	Reinhard	Pitié	PhotoWCT	Linear	WCT <sup>2</sup>	PhotoNAS	Bilateral	AdaCM
SSIM-Edge $\uparrow$	0.7631	0.7443	0.6315	0.7098	0.6758	0.7057	0.7765	0.7224
SSIM-Whole $\uparrow$	0.7746	0.7702	0.6623	0.7253	0.5987	0.7123	0.7766	0.7478
LPIPS $\downarrow$	0.1419	0.1556	0.2993	0.2360	0.2255	0.2135	0.1553	0.1780
Gram Loss $\downarrow$	8.0799	3.9796	3.6637	3.6781	3.6227	6.6976	3.3856	<b>3.1613</b>

Table 1: Quantitative evaluation results.

## Qualitative Results

In this section, we compare our proposed method against three state-of-the-art photo-realistic style transfer methods: Linear (Li et al. 2018a), WCT<sup>2</sup> (Yoo et al. 2019), and Bilateral (Xia et al. 2020) using default settings. Note that for Bilateral (Xia et al. 2020), we directly use stylization results provided by authors, since there are no available open-source codes. Given the limitation of length, we present more comparisons with PhotoWCT (Li et al. 2018b) and PhotoNAS (An et al. 2020) in the supplementary material.

We demonstrate several challenging cases in Fig. 5. Linear (Li et al. 2018a) applied spatial propagation network as post-processing step to reduce artifacts. However, as shown in Fig. 5, there are still noticeable distortions, including extra blurry texture and some dirty colors. WCT<sup>2</sup> achieves an excellent color transfer degree. However, it will generate hazy results in some cases (such as the 2-nd row). Meanwhile, when the color changes smoothly in a large region, WCT<sup>2</sup> may generate stylization results with stepped color bars (such as the 1-st, 3-rd and 7-th rows). This phenomenon may be caused by the whiten-and-color operation, since it occurs commonly in WCT-based methods. Bilateral performs quite well, but still has two types of artifacts. First, when the content or style image has bright light, Bilateral may generate results with overexposure regions (such as 2-nd, 7-th). Second, contrast reduction may appear in some cases, especially along strong edges. This artifact is also discussed in the original paper. Our proposed method achieves high-quality photo-realistic stylization, where local structures of content image are well preserved without distortions. Meanwhile, the color distribution of style image is properly transferred onto content image. In Fig. 4, we further demonstrate our stylization result on 4K resolution image.

## Quantitative Results

**Objective Metrics.** As the measurement of stylization tasks are usually subjective, we adopt some objective metrics for further comparison. Following previous works, between content images and output images, we adopt structural similarity (SSIM) index on whole images (SSIM-Whole) and their edge responses (SSIM-Edge) (Xie and Tu 2015) to measure detail preservation ability, and use LPIPS (Zhang et al. 2018) to measure perceptual similarity. Meanwhile, we adopt gram matrix difference (VGG style loss) between style images and output images to measure stylization effects.

All the evaluation is conducted on the dataset provided by Bilateral (Xia et al. 2020), and the results are shown in Table. 1. Classical methods Reinhard (Reinhard et al. 2001)

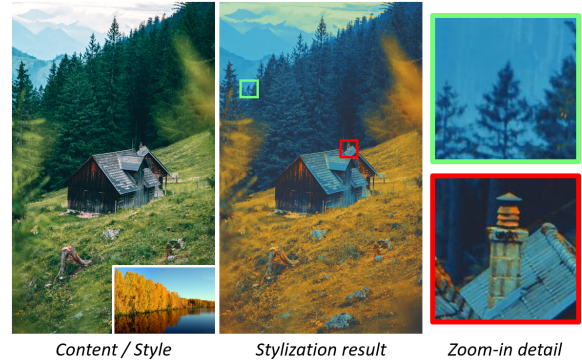


Figure 4: Visualization of 4K-resolution stylization.

Mean Score	Photorealism	Stylization	Overall
Linear	2.79	2.89	2.69
WCT <sup>2</sup>	2.92	3.21	2.97
PhotoNAS	2.51	2.59	2.41
Bilateral	3.66	3.50	3.54
Ours	<b>3.82</b>	<b>3.51</b>	<b>3.60</b>

Table 2: Results of user study. Higher score is better.

and Pitié (Pitie and Kokaram 2007) match the global statistics of pixels, focusing on the simple transfer like color shifts and tone curves. So the content details of their results tend to be well preserved, namely good photorealism. On the contrary, their gram loss is relatively high, for ignoring more complex style patterns. Among learning-based methods, our AdaCM achieves best stylization score. In terms of photorealism, AdaCM is slightly behind Bilateral. We argue this is due to the training instability of the hyper-parameter network paradigm and the lack of spatial context in ColorMLP. Besides, the photorealism scores are somewhat inconsistent with user study, because the subjective evaluation is more complex and abstract. To sum up, these results demonstrate the effectiveness of our AdaCM.

**User Study.** In this section, we conduct user study for subjective comparison. We recruit 30 users uncorrelated with the project. Following Bilateral (Xia et al. 2020), we randomly sample 15 content-style pairs and generate stylization results using Linear (Li et al. 2018a), WCT<sup>2</sup> (Yoo et al. 2019), PhotoNAS (An et al. 2020), Bilateral (Xia et al. 2020) and our AdaCM. For each pair, we display stylized images



Figure 5: Qualitative comparisons between our method and state-of-the-art photo-realistic stylization methods.

side-by-side in random order. And the users are asked to rate these results from 1 to 5 based on three aspects separately: photo-realistic (fewer artifacts), stylization degree (similar to style image), and overall quality. We totally collect 1,350 responses ( $15 \times 30 \times 3$ ). As shown in Table. 2, our method outperforms Linear, WCT<sup>2</sup> and PhotoNAS. Compared to Bilateral, our method has better performance on photo-realistic, similar performance on stylization degree and slightly better performance on overall quality.

**Efficiency Analysis.** We compare the efficiency of our method and other methods under multiple resolutions. The inference time of PhotoWCT, Linear, WCT<sup>2</sup>, and PhotoNAS are evaluated with official codes, and we directly refer the inference time of Bilateral (Xia et al. 2020) from the original paper since there is no open source code. As shown in Table. 3, our method is extremely fast and essentially invariant to practical resolutions. Our fast inference pipeline includes three steps: parameter prediction (4.79ms), generating 3D-LUT  $V_G$  using ColorMLP (0.26ms), and 3D-LUT based color manipulation. Thus, similar to (Xia et al. 2020), our method is extremely fast since the “heavy” encoder module of the network is fed with constant low resolution images ( $256 \times 256$ ). Meanwhile, results on Table. 3 demonstrate that ColorMLP is ultrafast under low resolution, but its computation cost grows rapidly with higher resolution. These results also demonstrate the necessity of using 3D-LUT as

an intermediary during inference. To conclude, our method achieves state-of-the-art photo-realistic stylization speed.

### Ablation Study

**Regularization.** We conduct ablation experiments to verify the effectiveness of each regularization term used during training. The results of two typical challenging cases are shown in Fig. 6, which demonstrates that: (1) Without using any regularization term, there are severe artifacts and distortions, especially in slow color transition region of original image such as sky. Meanwhile, the relative brightness relation in original image is broken, resulting in unnatural color distribution; (2) Adding regularization  $\mathcal{R}_{lut}$  on 3D-LUT, artifacts are partially removed but still obvious. This result indicates that adding smoothness regularization uniformly on full color space is insufficient; (3) Adding regularization  $\mathcal{R}_{img}$  only on stylization image, artifacts are largely removed and the relative brightness relation is well preserved. However, there are still artifacts in sky region; (4) Adding  $\mathcal{R}_{lut}$  and  $\mathcal{R}_{img}$  simultaneously, our full model can remove nearly all artifacts and generate photo-realistic stylization results. The final results indicate that  $\mathcal{R}_{lut}$  and  $\mathcal{R}_{img}$  are complementary during training AdaCM.

To further evaluate the effect of regularization, we visualize the color mapping relationship in ColorMLP with the help of 3D-LUT. As shown in Fig. 7, if no regularization

Method	Inference Time Per Image			
	512 <sup>2</sup>	1024 <sup>2</sup>	2000 <sup>2</sup>	4000 <sup>2</sup>
PhotoWCT	3.13s	3.32s	OOM	OOM
LST	1.32s	2.16s	OOM	OOM
WCT <sup>2</sup>	5.64s	9.41s	OOM	OOM
PhotoNAS	0.39s	0.60s	OOM	OOM
Bilateral*	<5ms	<5ms	<5ms	<5ms
ColorMLP	0.42ms	1.64ms	5.93ms	30.9ms
3D-LUT	0.06ms	0.09ms	0.32ms	1.09ms
Ours	5.11ms	5.14ms	5.37ms	6.14ms

Table 3: Efficiency comparison using an Nvidia V100 GPU (16GB RAM). OOM indicates out-of-memory. \* indicates the speed refers to its original paper. We evaluate the speed of ColorMLP with  $N = 20$  and 3D-LUT with  $M = 33$ . For AdaCM, the fast color manipulation strategy is used.



Figure 6: Ablation study of regularization term. There are no regularization terms used in baseline, and both  $\mathcal{R}_{lut}$  and  $\mathcal{R}_{img}$  are used in full model. Content loss  $\mathcal{L}_c$  and style loss  $\mathcal{L}_s$  are adopted in all experiments.

used, there are severe artifacts. Meanwhile, via generated 3D-LUT, we can find there exist drastic changes in color space, along with the broken of brightness monotonicity. With regularization, these artifacts are removed and we can obtain a smooth color mapping space.

**Number of Hidden Units.** We have discussed how many hidden units are enough for fitting a general 3D-LUT. In this section, we further conduct ablation experiments to verify how many hidden units are proper for photo-realistic stylization task. We train four AdaCM models with  $N = 5, 10, 20, 30$  separately, and show two challenging cases in Fig. 8, which demonstrate that: (1) with small number of hidden units (i.e.  $N = 5$ ), AdaCM can only model relatively simple color mapping relationship due to its limited capacity; (2) with large number of hidden units (i.e.  $N = 30$ ), there appears some severe artifacts in stylization results. We believe the main reason is that directly predicting large number of ColorMLP parameters is too difficult for the encoder. Based on these results, we find  $N = 20$  is a proper hidden units number for robust stylization results.

$M$ of 3D-LUT	8	16	33	64	128
PSNR	36.36	41.64	45.82	49.25	53.30

Table 4: Ablation study on the approximate ability of 3D-LUT with different  $M$ .

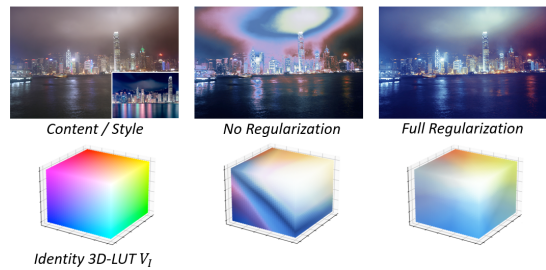


Figure 7: Visualization of generated ColorMLP with the help of 3D-LUT. Results of models trained with and without regularization term are both illustrated.

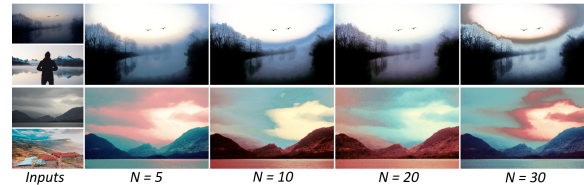


Figure 8: Ablation study of effects of number of hidden units used in ColorMLP.

**Number of Bins.** To improve inference efficiency in AdaCM, we design a fast pipeline with the help of 3D-LUT. The precision of transformation is determined by the number of bins  $M$ . 3D-LUT with larger  $M$  can achieve better approximation of ColorMLP. We conduct further ablation experiments to choose the proper value of  $M$ . We compute the average peak signal to noise ratio (PSNR) between stylized images directly generated by ColorMLP and generated by 3D-LUTs with different  $M$  on the evaluation dataset. As shown in Table. 4, PSNR becomes higher with the increasing of  $M$ . When  $M = 33$ , PSNR reaches 45.82, which is a high score and is hard for human to distinguish the differences. Meanwhile,  $M = 33$  is also a common choice empirically. Thus, 33 bins is a proper value in the fast inference pipeline.

## Conclusions

In conclusion, we propose a novel photo-realistic style transfer method AdaCM, which can synthesize high-resolution stylized image efficiently and effectively. In this work, we reveal the powerful ability of a lightweight MLP model to approximate complex non-linear color mapping relation. Then we utilize this remarkable property to formulate the photo-realistic stylization task as a simple MLP parameters prediction task. Experiments demonstrate that our method can generate high-quality and stable stylization results. Meanwhile, our methods can achieve real-time inference in 4K-resolution images. We believe the fast inference speed and the simple implementation of our method can enable more style transfer applications, especially in mobile devices. As for future work, we hope to further simplify the network structure to achieve faster speed, and explore better training constraint to achieve better stylization quality.

## References

- Affi, M.; Brubaker, M. A.; and Brown, M. S. 2021. Histogram: Controlling colors of gan-generated and real images via color histograms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7941–7950.
- An, J.; Xiong, H.; Huan, J.; and Luo, J. 2020. Ultrafast photorealistic style transfer via neural architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 10443–10450.
- Chen, J.; Adams, A.; Wadhwa, N.; and Hasinoff, S. W. 2016. Bilateral guided upsampling. *ACM Transactions on Graphics (TOG)*, 35(6): 1–8.
- Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2414–2423.
- Gharbi, M.; Chen, J.; Barron, J. T.; Hasinoff, S. W.; and Durand, F. 2017. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4): 1–12.
- Guo, C.; Li, C.; Guo, J.; Loy, C. C.; Hou, J.; Kwong, S.; and Cong, R. 2020. Zero-reference deep curve estimation for low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1780–1789.
- Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, 1501–1510.
- Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, 694–711. Springer.
- Kim, H.-U.; Koh, Y. J.; and Kim, C.-S. 2020a. Global and local enhancement networks for paired and unpaired image enhancement. In *European Conference on Computer Vision*, 339–354. Springer.
- Kim, H.-U.; Koh, Y. J.; and Kim, C.-S. 2020b. PieNet: Personalized Image Enhancement Network. In *European Conference on Computer Vision*, 374–390. Springer.
- Li, C.; and Wand, M. 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European conference on computer vision*, 702–716. Springer.
- Li, X.; Liu, S.; Kautz, J.; and Yang, M.-H. 2018a. Learning linear transformations for fast arbitrary style transfer. *arXiv preprint arXiv:1808.04537*.
- Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017. Universal style transfer via feature transforms. In *Advances in neural information processing systems*, 386–396.
- Li, Y.; Liu, M.-Y.; Li, X.; Yang, M.-H.; and Kautz, J. 2018b. A closed-form solution to photorealistic image stylization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 453–468.
- Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.
- Liu, S.; De Mello, S.; Gu, J.; Zhong, G.; Yang, M. H.; and Kautz, J. 2017. Learning affinity via spatial propagation networks. *Advances in Neural Information Processing Systems*, 2017: 1521–1531.
- Liu, Y.; He, J.; Chen, X.; Zhang, Z.; Zhao, H.; Dong, C.; and Qiao, Y. 2021. Very lightweight photo retouching network with conditional sequential modulation. *arXiv preprint arXiv:2104.06279*.
- Luan, F.; Paris, S.; Shechtman, E.; and Bala, K. 2017. Deep photo style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4990–4998.
- Park, D. Y.; and Lee, K. H. 2019. Arbitrary style transfer with style-attentional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5880–5888.
- Pitie, F.; and Kokaram, A. 2007. The linear Monge-Kantorovitch linear colour mapping for example-based colour transfer. In *4th European Conference on Visual Media Production*, 1–9. IET.
- Pitie, F.; Kokaram, A. C.; and Dahyot, R. 2005. N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, 1434–1439. IEEE.
- Reinhard, E.; Adhikhmin, M.; Gooch, B.; and Shirley, P. 2001. Color transfer between images. *IEEE Computer graphics and applications*, 21(5): 34–41.
- Sheng, L.; Lin, Z.; Shao, J.; and Wang, X. 2018. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8242–8250.
- Shih, Y.; Paris, S.; Barnes, C.; Freeman, W. T.; and Durand, F. 2014. Style transfer for headshot portraits. *ACM Transactions on Graphics (TOG)*, 33(4): 1–14.
- Shih, Y.; Paris, S.; Durand, F.; and Freeman, W. T. 2013. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics (TOG)*, 32(6): 1–11.
- Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.
- Song, Y.; Qian, H.; and Du, X. 2021. StarEnhancer: Learning Real-Time and Style-Aware Image Enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4126–4135.
- Tsai, Y.-H.; Shen, X.; Lin, Z.; Sunkavalli, K.; and Yang, M.-H. 2016. Sky is not the limit: semantic-aware sky replacement. *ACM Trans. Graph.*, 35(4): 149–1.
- Wu, F.; Dong, W.; Kong, Y.; Mei, X.; Paul, J.-C.; and Zhang, X. 2013. Content-based colour transfer. In *Computer Graphics Forum*, volume 32, 190–203. Wiley Online Library.

- Xia, X.; Zhang, M.; Xue, T.; Sun, Z.; Fang, H.; Kulis, B.; and Chen, J. 2020. Joint bilateral learning for real-time universal photorealistic style transfer. In *European Conference on Computer Vision*, 327–342. Springer.
- Xiao, X.; and Ma, L. 2009. Gradient-preserving color transfer. In *Computer Graphics Forum*, volume 28, 1879–1886. Wiley Online Library.
- Xie, S.; and Tu, Z. 2015. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, 1395–1403.
- Yoo, J.; Uh, Y.; Chun, S.; Kang, B.; and Ha, J.-W. 2019. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9036–9045.
- Zeng, H.; Cai, J.; Li, L.; Cao, Z.; and Zhang, L. 2020. Learning image-adaptive 3D lookup tables for high performance photo enhancement in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.