UCLA
UCLA Electronic Theses and Dissertations

Title
Learning and Investigating a Style-Free Representation for Fast, Flexible, and High-Quality Neural Style Transfer

Permalink
https://escholarship.org/uc/item/6w91d7ct

Author
Zhang, Chi

Publication Date
2019

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA
Los Angeles

Learning and Investigating a Style-Free Representation for Fast, Flexible, and High-Quality Neural Style Transfer

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Computer Science

by

Chi Zhang

2019

© Copyright by
Chi Zhang
2019

ABSTRACT OF THE THESIS

Learning and Investigating a Style-Free Representation for
Fast, Flexible, and High-Quality Neural Style Transfer

by

Chi Zhang
Master of Science in Computer Science
University of California, Los Angeles, 2019
Professor Song-Chun Zhu, Chair

We have just witnessed an unprecedented booming in the research area of artistic style transfer ever since Gatys *et al*. introduced the neural method. One of the remaining challenges is to balance a trade-off among three critical aspects—speed, flexibility, and quality: (i) the vanilla optimization-based algorithm produces impressive results for arbitrary styles, but is unsatisfyingly slow due to its iterative nature, (ii) the fast approximation methods based on feed-forward neural networks generate satisfactory artistic effects but bound to only a limited number of styles, and (iii) feature-matching methods like AdaIN achieve arbitrary style transfer in a real-time manner but at a cost of the compromised quality. We find it considerably difficult to balance the trade-off well by merely using a single feed-forward step and ask, instead, whether there exists an algorithm that could adapt quickly to any style, while the adapted model maintains high efficiency and good image quality. Motivated by this idea, we propose a novel method, coined *MetaStyle*, which formulates the neural style transfer as a bilevel optimization problem and combines learning with only a few post-processing update steps to adapt to a fast approximation model. The qualitative and quantitative analysis in the experiments demonstrates that the proposed approach achieves high-quality arbitrary artistic style transfer effectively, with a good trade-off among speed, flexibility, and quality. We also investigate the style-free representation learned by MetaStyle. Apart from style interpolation and video style transfer, we also implement well-known style transfer methods and examine the style transfer results after substituting the original content image inputs

with their style-free representation learned by MetaStyle. This could be thought of as inserting a preprocessing step to the content transformation branch. We show in the experiments that models trained using the MetaStyle preprocessing step produce consistently lower style loss and total loss, with a slightly higher content loss, compared to its counterparts without MetaStyle processing. And therefore, the stylized results achieve a better balance in appropriately combining semantics and styles. This shows that MetaStyle also learns a more general content representation in terms of adapting different artistic styles.

The thesis of Chi Zhang is approved.

Ying Nian Wu

Demetri Terzopoulos

Song-Chun Zhu, Committee Chair

University of California, Los Angeles

2019

To my family and Yuxin

who always brings me fun along the arduous journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I'd like to express my sincere gratitude towards my advisor Dr. Song-Chun Zhu for his guidance and mentorship and all the colleagues in VCLA, in particular, Yixin Zhu for showing me how proper research should be conducted, Feng Gao for our deep collaboration across numerous projects, Hangxin Liu for being an example of devotion, Xu Xie for helping me re-familiarize with `C++`, and Mark Edmonds for creating a welcoming working environment.

Last but not least, I'd like to thank my family members and my significant other, Yuxin Chi, for their continual emotional support for my study.

The work here is supported by the International Center for AI and Robot Autonomy (CARA).

CHAPTER 1

Introduction

To reduce the strenuous early-day efforts in producing pastiche, the computer vision and machine learning community have joined forces to devise automated algorithms to render a content image in the same style from a source artistic work. The style transfer problem covers a wide range of work, and at the beginning was phrased as a texture synthesis [DF81, ZWM98] problem. [EL99] first proposed to solve this problem by growing texture pixels one by one outward using a non-parametric sampling, and [WL00] accelerated this process by a tree-structured vector quantization. Patch-based sampling methods [EF01, LLX01] were later proposed to improve the synthesis quality and efficiency. [KEB05], however, viewed the problem from an energy minimization perspective and jointly optimized the objective using an EM-like algorithm. The concept of image analogies [HJO01] was also introduced to produce the "filtered" results and later extended by [ZZ11] to tailor to portrait paintings.

With the recent boost of deep neural networks and large datasets in computer vision, [GEB16] first discovered that combining multi-level VGG features [SZ14] trained on the Im-

Figure 1.1: Style transfer results using MetaStyle, balancing the three-way trade-off among speed, flexibility, and quality. Left: the content image and the style-free representation learned by MetaStyle. Right: the stylized images from 14 different styles.

Method	Speed	Flexibility	Quality	Drawback
Optimization-based	Slow	Any	High	Run for each content-style pair
Fast approximation	Fast	Single	High	Train long for each new style
Feature-matching	Fast	Any/Several	Compromised	Limited set of styles, low quality

Table 1.1: Pros and cons of existing neural style transfer methods in the three metrics: speed, flexibility, and quality.

ageNet [DDS09] successfully captured the characteristics of the style while balancing the statistics of the content, producing impressive results for the task of artistic style transfer. This serendipitous finding has brought to life a surge of interests in the research area of style transfer. Iterative optimization methods [GEB15, GEB16, LW16] generate artistic images that well interpolate between arbitrary style space and content space; but due to its iterative nature, these methods are generally slow, requiring hundreds of update steps for each content-style pair and becoming impractical for deployment in products. Fast approximation methods using feed-forward neural networks trained with *perceptual loss* [JAF16, DSK17, ZD17] overcome the speed problem and usually result in satisfactory artistic effects; however, good quality is limited to a single or a small number of style images, sacrificing the flexibility in the original method. Feature-matching methods [HB17, SLS18] achieve arbitrary style transfer in real-time, but these models come at the cost of compromised style transfer quality, compared to the methods mentioned above. Table 1.1 summarizes the pros and cons of existing methods in the field.

To address these problems, we argue that it is nontrivial to use either sheer iterative optimization methods or single-step feed-forward approximations to achieve the three-way trade-off among speed, flexibility, and quality. In this work, we seek to find, instead, an algorithm that would fast adapt to any style by a small or even negligible number of post-processing update steps, so that the adapted model keeps high efficiency and satisfactory generation quality.

Specifically, we propose a novel style transfer algorithm, coined *MetaStyle*, which formulates the fast adaptation requirement as the bilevel optimization, solvable by the recent

meta-learning methods [FAL17, NAS18]. This unique problem formulation encourages the model to learn a style-free representation for content images, and to produce a new feed-forward model, after only a small number of update steps, to generate high-quality style transfer images for a single style efficiently. From another perspective, this formulation could also be thought of as finding a style-neutral input for the vanilla optimization-based methods [GEB16], but transferring styles much more effectively.

Our model is instantiated using a neural network. The network structure is inspired by the finding [DSK17] that scaling and shifting parameters in instance normalization layers [UVL17] are specialized for specific styles. In contrast, unlike prior work, our method implicitly forces the parameters to find no-style features in order to rapidly adapt and remain parsimonious in terms of the model size. The trained MetaStyle model has roughly the same number of parameters as described in [JAF16], and requires merely 0.1 million training steps.

Comprehensive experiments with both qualitative and quantitative analysis, compared with prior neural style transfer methods, demonstrate that the proposed method achieves a good trade-off among speed, flexibility, and quality. Figure 1.1 shows sample results using the proposed style transfer.

Apart from the proposed MetaStyle algorithm, we also investigate the representation learned by MetaStyle. Specifically, we investigate the learned representation in tasks of style interpolation and video style transfer. We also consider substituting the content image inputs in famous arbitrary style transfer methods (*e.g.*, [ZD17] and [GLK17]) with the style-free representation obtained after running the MetaStyle model. During retraining of these models, we notice that the style loss and the total loss of MetaStyle-processed models are consistently lower than their counterparts', though the content loss becomes slightly higher. Images stylized using the MetaStyle extention also show a better balance between semantics and styles. These effects demonstrate the generalizability of the learned representation from MetaStyle.

The contributions of the work could be summarized as follows:

- We propose a new style transfer method called MetaStyle to achieve the three-way

trade-off in speed, flexibility, and quality. To the best of our knowledge, this is the first work that formulates the style transfer as the bilevel optimization so that the model could be easily adapted to a new style with only a small number of updates, producing high-quality results while remaining parsimonious.

- The proposed method provides a style-free representation, from which a fast feed-forward high-quality style transfer model could be adapted after only a small number of iterations, making the cost of training a high-quality model for a new style nearly negligible.

- We also investigate the learned representation of MetaStyle and show in the experiments that models trained with a MetaStyle preprocessing step show consistently lower style loss and total loss, with better stylized results.

CHAPTER 2

Related Work

2.1 Neural Style Transfer

By leveraging the pre-trained VGG model [SZ14], [GEB16] first proposed to explicitly separate content and style: the model has a feature-matching loss involving the second-order Gram matrices (later called *perceptual loss*) and iteratively updates the input images (usually hundreds of iterations) to produce high-quality style transfer results. To overcome the speed limit, [JAF16] recruited an image transformation network to generate stylized results sufficiently close to the optimum solution directly. Concurrent work by [ULV16] instantiated a similar idea using multi-resolution generator network and further improved the diversity of the generated images [UVL17] by applying the Julesz ensembles [ZWM98, ZLW00]. Note that each trained model using any of these methods is specialized to a single style.

Significant efforts have been made to improve the neural style transfer. [LW16] modeled the process using an Markov random field (MRF) and introduced the MRF loss for the task. [LWL17] discovered that the training loss could be cast in the maximum mean discrepancy framework and derived several other loss functions to optimize the content image. [CYL17] jointly learned a style bank for each style during model training. [DSK17] modified the instance normalization layer [UVL17] to condition on each style. [ZD17] proposed to use a CoMatch layer to match the second-order statistics to ease the learning process. Although these approaches produce transfer results of good quality in real-time for a constrained set of styles, they still lack the generalization ability to transfer to arbitrary styles. Additionally, these approaches sometimes introduce additional parameters proportional to the number of the styles they learn.

Recent work concentrated on more generalizable approaches. A patch-based style swap layer was first introduced [CS16] to replace the content feature patch with the closest-matching style feature patch, and a compromised inverse network was employed for fast approximation. The adaptive instance normalization layer [HB17] was introduced to scale and shift the normalized content features by style feature statistics and act as the bottleneck in an encoder-decoder architecture, while similarly [LFY17] applied recursive whitening and coloring transformation in multi-level pre-trained auto-encoder architecture. More recent works include a ZCA-like style decorator and an hourglass network that were integrated in a multi-scale manner [SLS18] and a meta network that was trained to generate parameters of an image transformation network [SYZ18] directly. These methods, though efficient and flexible, often suffer from compromised image generation quality, especially for the unobserved styles. In contrast, the proposed model could adapt to any style quickly without sacrificing the speed or the image quality, making its final performance on par with fast approximation methods, *e.g.*, [JAF16].

Additionally, our model is also parsimonious, requiring roughly the same number of model parameters as [JAF16], using merely 0.1 million iterations. In comparisons, *e.g.*, [GLK17] extended the conditional instance normalization framework [DSK17], but required a pre-trained Inception-v3 [SVI16] to predict the parameters for a single style. This model requires 4 million update steps, making training burdensome.

2.2 Meta-Learning

Meta-learning has been successfully applied in few-shot learning with early work dated back to the 1990's. Here we only review one branch focusing on *initialization strategy* [FFS18] that influences our work. [RL16] first employed an LSTM network as a meta-learner to learn an optimization procedure. [FAL17] proposed model-agnostic meta-learning (MAML) so that a model previously learned on a variety of tasks could be quickly adapted to a new one. This method, however, required second-order gradient computation in order to derive gradient for the meta-objective correctly, and therefore consumed significant computational

power, though a first-order method was also tested with compromised performance.

Following their work, [NAS18] generalized MAML to a family of algorithms and extended it to Reptile. Reptile coupled sequential first-order gradients with advanced optimizers, such as Adam [KB14], resulting in an easier implementation, shorter training time and comparable performance. A recent work [SYZ18] modeled the process of neural style transfer using an additional large group of fully-connected layers such that the parameters of an image transformation network could be predicted. In contrast, the proposed method remains parsimonious with a single set of parameters to train and adapt.

As we will show in the Section 4.1, the meta network is, *de facto*, a special case in the proposed bilevel optimization framework. To the best of our knowledge, our work is the first to explicitly cast neural style transfer as the bilevel optimization problem in the initialization strategy branch.

CHAPTER 3

Background

Before detailing the proposed model, we first introduce two essential building blocks, *i.e.*, the perceptual loss and the general bilevel optimization problem, which lay the foundation of the proposed approach.

3.1 Style Transfer and Perceptual Loss

Given an image pair (I_c, I_s), the style transfer task aims to find an "optimal" solution I_x that preserves the content of I_c in the style of I_s. [GEB16] proposed to measure the optimality with a newly defined loss using the trained VGG features, later modified and named as the perceptual loss [JAF16]. The perceptual loss could be decomposed into two parts: the content loss and the style loss.

Denoting the VGG features at layer i as $\phi_i(\cdot)$, the content loss $\ell_{\text{content}}(I_c, I_x)$ is defined using the L_2 norm

$$\ell_{\text{content}}(I_c, I_x) = \frac{1}{N_i} \left\| \phi_i(I_c) - \phi_i(I_x) \right\|_2^2 , \tag{3.1}$$

where N_i denotes the number of features at layer i.

The style loss $\ell_{\text{style}}(I_s, I_x)$ is the sum of Frobenius norms between the Gram matrices of the VGG features at different layers

$$\ell_{\text{style}}(I_s, I_x) = \sum_{i \in S} \left\| G(\phi_i(I_s)) - G(\phi_i(I_x)) \right\|_F^2 , \tag{3.2}$$

where S denotes a predefined set of layers and G the Gramian transformation.

The transformation could be efficiently computed by

$$G(x) = \frac{\psi(x)\psi(x)^T}{CHW} \tag{3.3}$$

for a 3D tensor x of shape $C \times H \times W$, where $\psi(\cdot)$ reshapes x into $C \times HW$.

The perceptual loss $\ell(I_c, I_s, I_x)$ aggregates the two components by the weighted sum

$$\ell(I_c, I_s, I_x) = \alpha \ell_{\text{content}}(I_c, I_x) + \beta \ell_{\text{style}}(I_s, I_x). \tag{3.4}$$

3.2 Bilevel Optimization

We formulate the style transfer problem as the bilevel optimization in the form simplified by [FFS18]

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & E(w_\theta, \theta) \\ \text{subject to} \quad & w_\theta = \arg\min_w L_\theta(w), \end{aligned} \tag{3.5}$$

where E is the *outer objective* and L_θ the *inner objective*. Under differentiable L_θ, the constraint could be replaced with $\nabla L_\theta = 0$. However, in general, no closed-form solution of w_θ exists and a practical approach to approximate the optimal solution is to replace the inner problem with the gradient dynamics, *i.e.*,

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & E(w_T, \theta) \\ \text{subject to} \quad & w_0 = \Psi(\theta) \\ & w_t = w_{t-1} - \delta \nabla L_\theta(w_{t-1}) \end{aligned} \tag{3.6}$$

where Ψ initializes w_0, δ is the step size and T the maximum number of steps. [FFS18] proved the convergence of Equation 3.6 under certain conditions. Though they did not model their problems using bilevel optimization but rather an intuitive motivation, [FAL17] and [NAS18] both use the identity mapping for Ψ, with the former computing the full gradient for θ to optimize the outer objective, and the latter one only the first-order approximate gradient.

CHAPTER 4

MetaStyle

In this section, we first detail the intuition behind and the formulation of the proposed framework, explain the design choices and discuss relations to the previous approaches. Then the network architecture is presented with the training protocol and the detailed algorithm. Finally, we discuss how we investigate the representation learned by MetaStyle.

4.1 Problem Formulation

MetaStyle is tasked with finding a three-way trade-off among speed, flexibility, and quality in neural style transfer. To achieve such a balance, however, we argue that it is nontrivial to either merely use iterative optimization methods or simply adopt single-step feed-forward approximations. To address this challenge, we consider a new approach where we first learn a style-neutral representation and allow a limited number of update steps to this neutral representation in the post-processing stage to adapt to a new style. It is expected that the model should generate a stylized image efficiently after adaptation, be general enough to accommodate any new style, and produce high-quality results.

To this end, we employ an image transformation network with content image input [JAF16] and cast the entire neural style transfer problem in a bilevel optimization framework [FFS18]. As discussed in Equation 3.6, we choose to model θ as the network initialization and w_T the adapted parameters, now denoted as $w_{s,T}$, to emphasize the style to adapt to. T is restricted to be small, usually in the range between 1 and 5. Both the inner and outer objective is designed to be the perceptual loss averaged across datasets. However, as described in meta-learning [FAL17, NAS18], the inner objective uses a model initialized

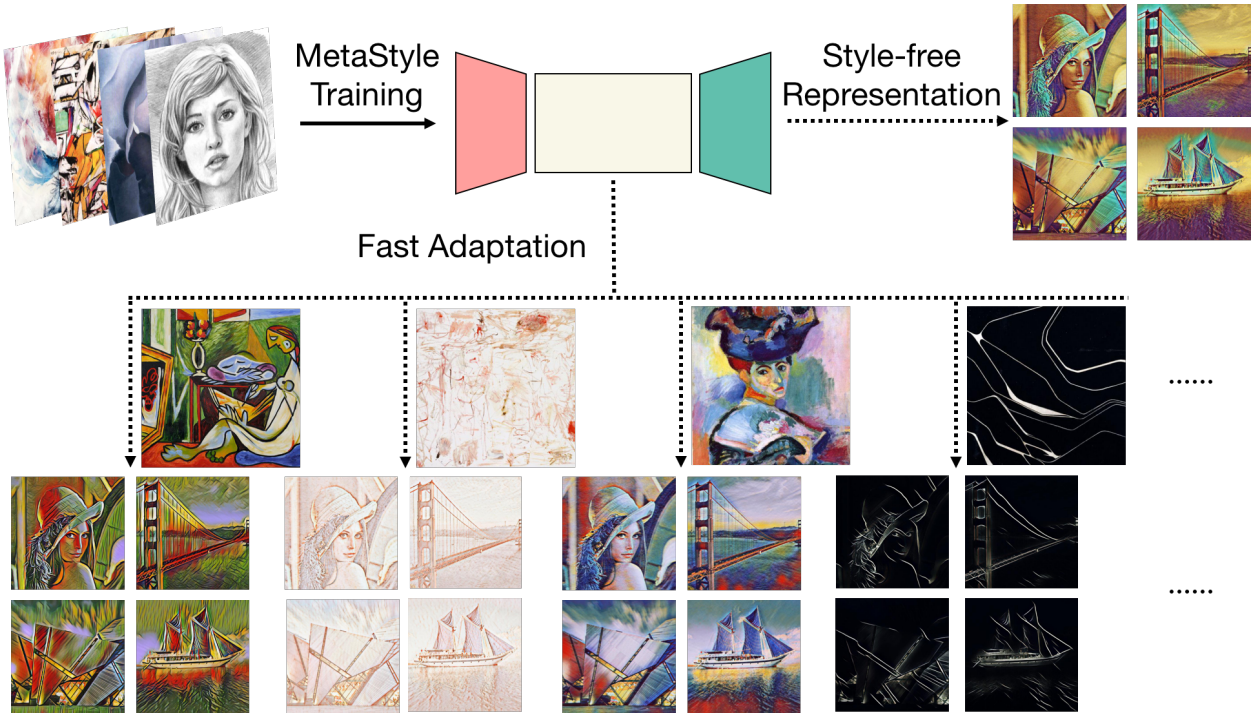

Figure 4.1: The proposed MetaStyle framework, in which the model is optimized using the bilevel optimization over large-scale content and style dataset. The framework first learns a style-neutral representation. A limited number of post-processing update steps is then applied to adapt the model quickly to a new style. After adaptation, the new model serves as an image transformation network with good transfer quality and high efficiency.

with θ and only optimizes contents in the training set, whereas the outer objective tries to generalize to contents in the validation set. Ψ is the identity mapping. Formally, the problem could be stated as

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \mathbb{E}_{c,s}[\ell(I_c, I_s, M(I_c; w_{s,T}))] \\ \text{subject to} \quad & w_{s,0} = \theta \\ & w_{s,t} = w_{s,t-1} - \delta \nabla \mathbb{E}_c[\ell(I_c, I_s, M(I_c; w_{s,t-1}))], \end{aligned} \tag{4.1}$$

where $M(\cdot;\cdot)$ denotes our model and δ the learning rate of the inner objective. The expectation of the outer objective $\mathbb{E}_{c,s}$ is taken with respect to both the styles and the content images in the validation set, whereas the expectation of the inner objective $\mathbb{E}_c$ is taken with respect to the content images in the training set only. This design allows the adapted model

to specialize for a single style but still maintain the initialization generalized enough. Note that for the outer objective, $w_{s,T}$ implicitly depends on θ. In essence, the framework learns an initialization $M(\cdot;\theta)$ that could adapt to $M(\cdot;w_{s,T})$ efficiently and preserve high image quality for an arbitrary style. Figure 4.1 shows the proposed framework.

The explicit training-validation separation in the framework forces the style transfer model to generalize to unobserved content images without over-fitting to the training set. Coupled with this separation, MetaStyle constrains the number of steps in the gradient dynamics computation to encourage quick adaptation for an arbitrary style and, at the same time, picks an image transformation network due to its efficiency and high transfer quality. These characters serve to the trade-offs among speed, flexibility, and quality.

We now discuss MetaStyle's relations to other methods.

4.1.1 Relation to Johnson *et al.* [JAF16]

Johnson *et al.*'s method finds an image transformation model tailored to a single given style, minimizing the model parameters by

$$\underset{w}{\text{minimize}} \quad \mathbb{E}_c[\ell(I_c, I_s, M(I_c; w))], \tag{4.2}$$

where the expectation is taken with respect to only the contents. In contrast, in Equation 4.1, we seek a specific model *initialization* θ, which is not the final parameters used for the style transfer, but could adapt to any other style using merely a small number of post-processing updates. Assuming there exists an implicit, unobserved neutral style, MetaStyle could be regarded as learning a style-free image transformation. Only after learning is complete, do we use Equation 4.2 to quickly adapt the initialization to any style.

4.1.2 Relation to Gatys *et al.* [GEB16]

Starting with the content image, Gatys *et al.* finds the minimizer of the perceptual loss using iterative updates. From this iterative update perspective, MetaStyle could be regarded as learning to find a good starting point for the optimization algorithm. This learned transfor-

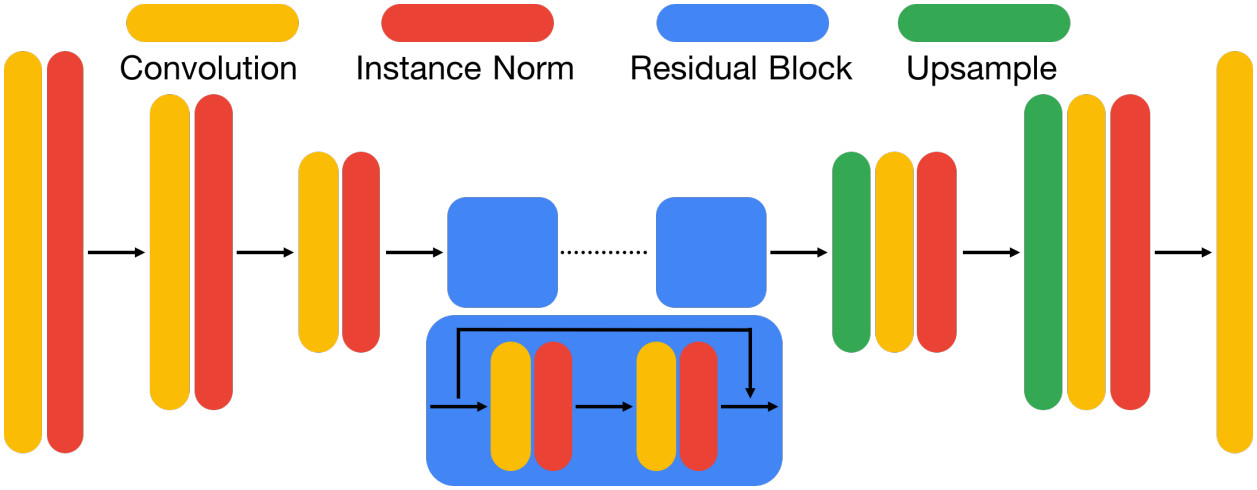

Figure 4.2: Network architecture. Residual Blocks are stacked multiple times to extract deeper image features.

mation generates a style-neutral image while dramatically reducing the number of update steps.

4.1.3 Relation to Shen *et al.* [SYZ18]

Shen *et al.*'s method is a special case of the proposed bilevel optimization framework, where $T = 0$ and Ψ is a highly nonlinear transformation, parameterized by θ, that uses a style image to predict parameters of another image transformation network.

4.2 Network Architecture, Training & Algorithm

Our network architecture largely follows that of an image transformation network described in [DSK17]. However, unlike the original architecture, the output of the last convolution layer is unnormalized and activated using the Sigmoid function to squash it into $[0, 1]$. Upsampled convolution, which first upsamples the input and then performs convolution, and reflection padding are used to avoid checkerboard effects [ZD17]. Inspired by the finding [DSK17] that scaling and shifting parameters in the instance normalization layers specialize for specific styles, we append an instance normalization layer after each convolution layer, except the

Algorithm 1: MetaStyle

Input : content training dataset $\mathcal{D}_{tr}$, content validation dataset $\mathcal{D}_{val}$, style dataset $\mathcal{D}_{style}$, inner learning rate δ, outer learning rate η, number of inner updates T

Output: trained parameters θ

```
randomly initialize θ
while not done do
    initialize outer loss E ← 0
    sample a batch of styles from D_style
    for each style I_s do
        w_s ← θ
        for i ← 1 to T do
            sample a batch B_tr from D_tr
            compute inner loss L_θ using I_s and B_tr
            w_s ← w_s − δ∇L_θ
        end
        sample a batch B_val from D_val
        increment E by loss from I_s and B_val
    end
    θ ← θ − η∇E
end
```

last. See Figure 4.2 for a graphical illustration. This design forces the parameters in instance normalization layers to learn from an implicit, unobserved neutral style while keeping the model size parsimonious. Table 4.1 provides detailed network architecture design. Note that all the convolution layers use the "same" padding before the operation, and all the upsamplings are of nearest sampling with a scale factor of 2.

For training, we use small-batch learning to approximate both the inner and outer objective. The inner objective is approximated by several batches sampled from the training dataset and computed on a single style, whereas the outer objective is approximated by a style batch, in which each style incurs a perceptual loss computed over a content batch sampled from the validation dataset. The problem is solvable by MAML [FAL17] and summarized in Algorithm 1. After training, θ could be used as the initialization to minimize

Operator	Channel	Stride	Kernel	Padding	Activation
Network — Input	3				
Convolution	32	1	9	Reflection	
Instance Norm	32				ReLU
Convolution	64	2	3	Reflection	
Instance Norm	64				ReLU
Convolution	128	2	3	Reflection	
Instance Norm	128				ReLU
Residual Block	128				
Residual Block	128				
Residual Block	128				
Residual Block	128				
Residual Block	128				
Upsampling					
Convolution	64	1	3	Reflection	
Instance Norm	64				ReLU
Upsampling					
Convolution	32	1	3	Reflection	
Instance Norm	32				ReLU
Convolution	3	1	9	Reflection	Sigmoid
Residual Block — Input	128				
Convolution	128	1	3	Reflection	
Instance Norm	128				ReLU
Convolution	128	1	3	Reflection	
Instance Norm	128				
Addition	128				

Table 4.1: Network architecture used in MetaStyle.

Equation 4.2 to adapt the model to a single style or to provide the starting point $M(I_c;\theta)$ for optimization-based methods.

During training, we use the time-based learning rate decay for both the outer and the inner objective optimization, *i.e.*,

$$\kappa = \frac{1}{1+k\times t}\kappa_0, \tag{4.3}$$

where κ denotes the learning rate for either the outer or the inner objective, t the number of iterations, κ_0 the initial learning rate, and $k = 2.5\times 10^{-5}$. To reduce the computation, we do not iteratively sample a new content batch from the training set $\mathcal{D}_{tr}$ in the inner objective optimization, but share the same content batch $\mathcal{B}_{tr}$ in each iteration. Similarly, we use the same content batch $\mathcal{B}_{val}$ from the validation set $\mathcal{D}_{val}$ during each outer objective update. Note that this procedure accelerates the convergence. In contrast to [FAL17] and [NAS18], we find that the first-order gradient approximations lead to serious fluctuations during training and no convergence is observed. In addition, increasing T to the values as large as 5 does not notably improve performance. Therefore, we set $T = 1$ in the reported experiment results. Such a setting encourages the model to fast adapt to any style and also significantly reduces GPU memory consumption. To further stabilize training, we only update parameters in instance normalization layers in inner objective optimization. This design implicitly encourages the instance normalization layers to find a set of parameters that specializes in a style-neutral representation, corresponding to the finding in [DSK17].

4.3 Investigating the Style-Neutral representation

Experimental settings of style interpolation and video style transfer have been detailed in [HB17, SLS18], and here we only discuss how MetaStyle could be used as a preprocessing step.

As mentioned above, our model $M(\cdot;\theta)$ initialized with θ could be generally regarded as one that strips off the styles in the content images, *i.e.*, one that produces a style-free representation while preserving the semantic structures in the original images. This

design introduces new possibilities in investigating style transfer methods in the sense that MetaStyle representation could be swapped in into any style transfer methods, *e.g.*, [GEB16, JAF16, LFY17, HB17, GLK17] as a preprocessing step in the content transformation branch. Specifically, MetaStyle preprocesses each content image such that its original style is stripped off and transferring a new style becomes easier. As an example for the arbitrary style transfer setting, the problem could be formulated as

$$\underset{w}{\text{minimize}} \quad \mathbb{E}_{c,s}[\ell(I_c, I_s, M(\text{MetaStyle}(I_c; \theta), I_s; w)], \tag{4.4}$$

where we explicitly separate notations for our MetaStyle model and general style transfer model $M(\cdot, \cdot; \cdot)$. The formulation is similar for the vanilla optimization setting and the fast approximation setting.

CHAPTER 5

Experiments

5.1 Implementation Details

We train our model using MS-COCO [LMB14] as our content dataset and WikiArt test set [Nic16] as our style dataset. The content dataset has roughly 80,000 images and the WikiArt test set 20,000 images. We use Adam [KB14] with a learning rate 0.001 to optimize the outer objective and vanilla SGD with a learning rate 0.0001 for the inner objective. All batches are of size 4. We fix $\alpha = 1$, $\beta = 1 \times 10^5$ across all the experiments. Content loss is computed on `relu2_2` of a pre-trained VGG16 model and style loss over `relu1_2`, `relu2_2`, `relu3_3` and `relu4_3`. The entire model is trained on a Nvidia Titan Xp with only 0.1 million iterations.

For investigation into MetaStyle representation, apart from style interpolation and video style transfer, we retrain the vanilla optimization-based style transfer algorithm [GEB16], the fast approximation method [JAF16], and Google's predicted conditional instance normalization (PCIN) [GLK17], except that the content image inputs are preprocessed by the trained MetaStyle representation, *i.e.*, MetaStyle$(I_c; \theta)$. The hyperparameters of these models are fine-tuned on the MS-COCO and WikiArt datasets.

5.2 Comparison with Prior Arts

We compare the proposed MetaStyle with existing methods [GEB16, JAF16, LFY17, HB17, SYZ18, SLS18, CS16] in terms of speed, flexibility, and quality. Specifically, for these existing methods, we use the pre-trained models made publicly available by the authors. To adapt MetaStyle to a specific style, we train the MetaStyle model using only 200 iterations on MS-

COCO dataset, which amounts to an additional 24 seconds of training time with a Titan Xp GPU. For Gatys *et al.*, we optimize the input using 800 update steps. For Chen *et al.*, we use its fast approximation model. All five levels of encoders and decoders are employed in our experiments involving Li *et al.*.

5.2.1 Speed and Flexibility

Table 5.1 summarizes the benchmarking results regarding style transfer speed and model flexibility. As shown in the table, our method achieves the same efficiency as Johnson *et al.* and Shen *et al.*. Additionally, unlike Shen *et al.* that introduces a gigantic parameter prediction model, MetaStyle is parsimonious with roughly the same number of parameters as Johnson *et al.*. While Johnson *et al.* requires training a new style model from scratch, MetaStyle could be immediately adapted to any style with a negligible number of updates under 30 seconds. This property significantly reduces the efforts in arbitrary style transfer and, at the same time, maintains a high image generation quality, as shown in the next

Method	Param	256 (s)	512 (s)	# Styles
Gatys *et al.*	N/A	7.7428	27.0517	∞
Johnson *et al.*	**1.68M**	**0.0044**	**0.0146**	1
Li *et al.*	34.23M	0.6887	1.2335	∞
Huang *et al.*	7.01M	0.0165	0.0320	∞
Shen *et al.*	219.32M	**0.0045**	**0.0147**	∞
Sheng *et al.*	147.22M	0.5089	0.6088	∞
Chen *et al.*	**1.48M**	0.2679	1.0890	∞
Ours	**1.68M**	**0.0047**	**0.0145**	$\infty^\star$

Table 5.1: Speed and flexibility benchmarking results. Param lists the number of parameters in each model. 256/512 denotes inputs of 256×256/512×512. # Styles represents the number of styles a model could potentially handle. ⋆Note that MetaStyle adapts to a specific style after very few update steps and the speed is measured for models adapted.

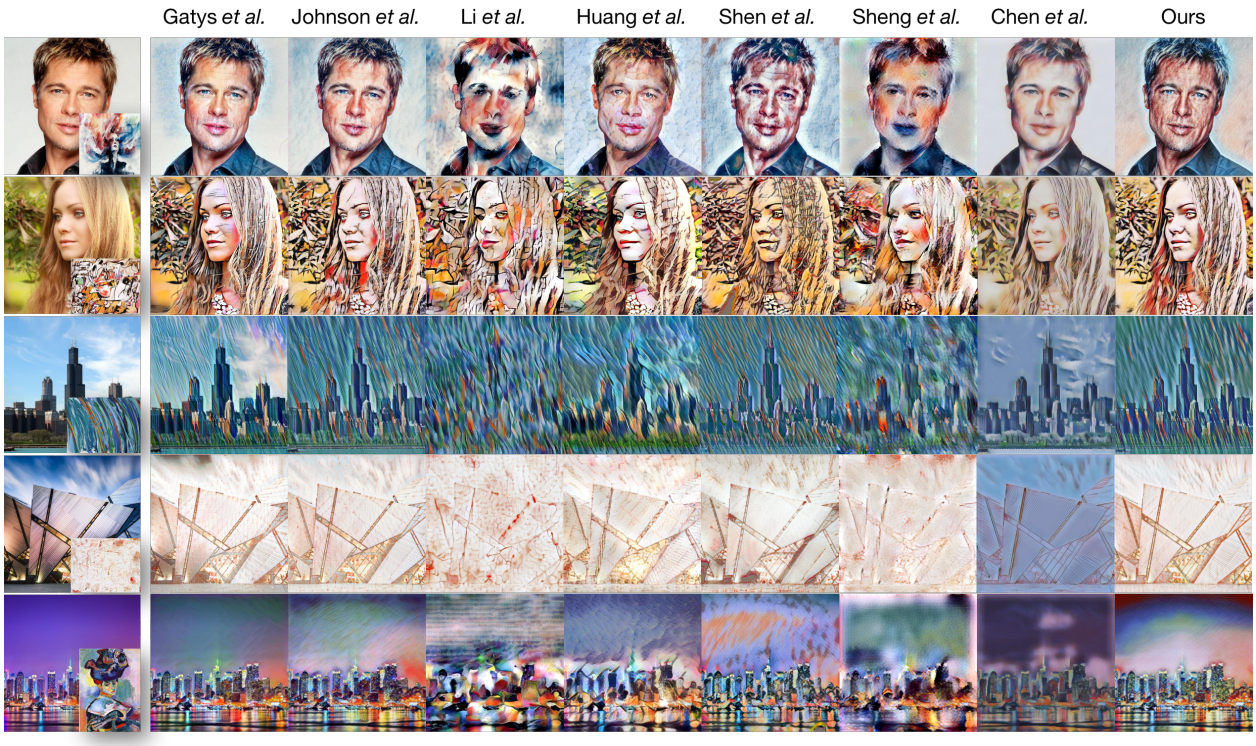

Figure 5.1: Qualitative comparisons of neural style transfer between the existing methods and the proposed MetaStyle using bilevel optimization. Arbitrary style transfer models observe neither the content images nor the style images during training.

paragraph.

5.2.2 Quality

Figure 5.1 shows the qualitative comparisons of the style transfer between the existing methods and the proposed MetaStyle method. We notice that, overall, Gatys *et al*. and Johnson *et al*. obtain the best image quality among all the methods we tested. This observation coheres with our expectation, as Gatys *et al*. iteratively refines a single input image using an optimization method, whereas the model from Johnson *et al*. learns to approximate optimal solutions after seeing a large number of images and a fixed style, resulting in a better generalization.

Among methods capable of arbitrary style transfer, Li *et al*. applies style strokes ex-

(a) Two-style interpolation results. The content image and style images are shown on the two ends.

(b) Video style transfer results. The left pane shows the style and the right pane contents and stylized video sequence.

Figure 5.2: Style interpolation and video style transfer.

cessively to the contents, making the style transfer results become deformed blobs of color, losing much of the image structures in the content images. Looking deep into Huang *et al.*, we notice that the arbitrary style transfer method produces images with unnatural cracks and discontinuities. Results from Shen *et al.* come with strange and peculiar color regions that likely result from non-converged image transformation models. Sheng *et al.* unnecessarily morphs the contours of the content images, making the generated artistic effects inferior. The inverse network from Chen *et al.* seems to apply the color distribution in the style image to the content image without successfully transferring the strokes and artistic effects in style.

In contrast, MetaStyle achieves a right balance between styles and contents comparable to Johnson *et al.*. Such property should be attributed to the image transformation network shown in Johnson *et al.*, while the fast adaptation comes from our novel formulation.

For more examples stylized by MetaStyle, please refer to the supplementary file of [ZZZ18].

5.3 Investigating the MetaStyle representation

In this section, we first present the performance of MetaStyle on style interpolation and video style transfer to show the generalizability of the representation, and then we quantitatively

and qualitatively compare performance of previous methods with and without MetaStyle preprocessing.

5.3.1 Style Interpolation

To interpolate among a set of styles, we perform a convex combination on the parameters of adapted MetaStyle models learned after 200 iterations using

$$\begin{aligned} w &= \sum_{s_i \in S} \gamma_i w_{s_i} \\ \text{subject to} \quad & \sum_i \gamma_i = 1 \\ & \forall i, \gamma_i \geq 0, \end{aligned} \tag{5.1}$$

where S denotes the style set, s_i a specific style in the set, w the adapted parameters, and γ the weighting coefficient. Figure 5.2a shows the results of a two-style interpolation.

5.3.2 Video style transfer

We perform the video style transfer by first training the MetaStyle model for 200 iterations to adapt to a specific style, and then applying the transformation to a video sequence frame by frame. Figure 5.2b shows the video style transfer results in five consecutive frames. Note that our method does not introduce the flickering effect that harms aesthetics.

5.3.3 MetaStyle as Preprocessing for Gatys *et al.* [GEB16]

As mentioned in Section 4.1, MetaStyle, before adaptation, provides a style-neutral representation and serves as a better starting point for the optimization-based method. Also, transforming the content inputs using MetaStyle before optimization could be more broadly regarded as preprocessing. We empirically illustrate in Figure 5.3, in which we compare initializing the optimization with either the content image or the style-neutral representation. We notice that after 150 steps, Gatys *et al.* only starts to apply minor style strokes to the content while MetaStyle-initialized method could already produce a well-stylized re-

sult. Given that MetaStyle is not directly formulated to find a good starting point, this effect is surprising, showing the generalization ability of the representation discovered by the proposed MetaStyle.

5.3.4 MetaStyle as Preprocessing for Johnson *et al.* [JAF16]

Similar to the previous section, we could directly insert MetaStyle before optimizing the fast approximation model in Johnson *et al.* [JAF16], and treat it as a preprocessing step. However, due to the extreme similarity of the network architectures used in MetaStyle and [JAF16] and the training objectives in MetaStyle's adaptation step and the training step in [JAF16], one simpler strategy is to directly initialize the network using MetaStyle's learned parameters and optimize the objective thereafter. Figure 5.4 shows the results after 200 training iterations and the curve for the perceptual loss during evaluation. It is evident that while Johnson *et al.* still struggles to figure out a well-balanced interpolation between the style

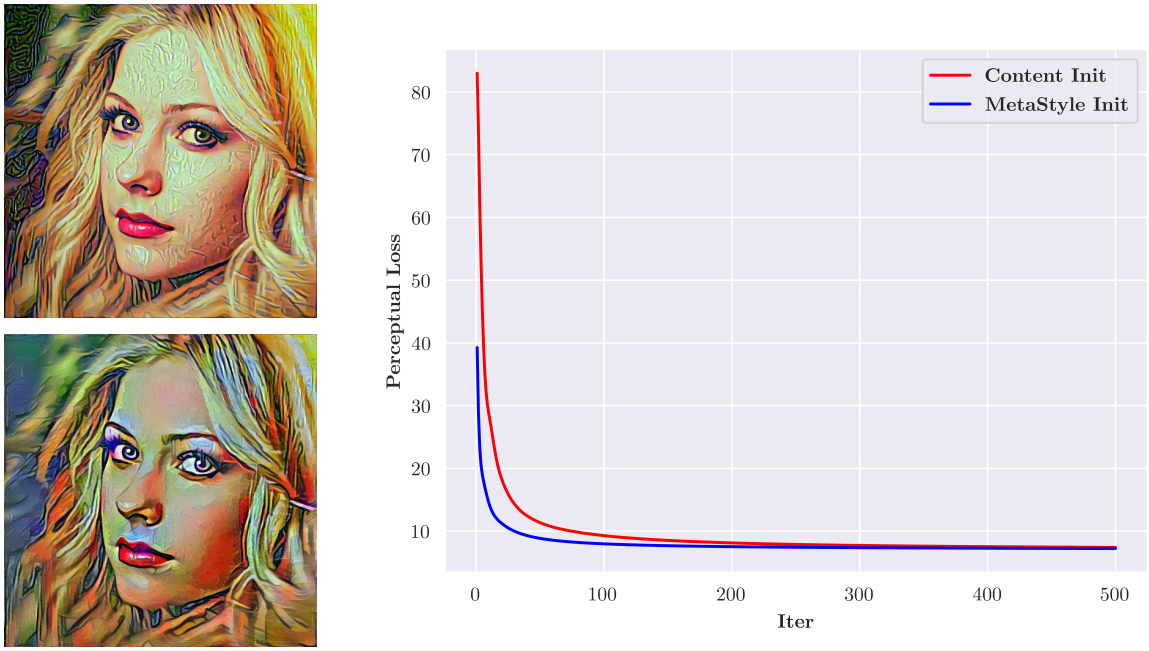

Figure 5.3: Comparison with Gatys *et al.*. (Left) The results using (upper) Gatys *et al.* and (lower) the proposed MetaStyle. (Right) The perceptual loss.

manifold and the content manifold, MetaStyle could already generate a high-quality style transfer result with a good equilibrium between style and content. This contrast becomes even more significant considering that a fully trained model from Johnson *et al.* requires about 160,000 iterations and an adapted MetaStyle model only 200. The loss curve also shows consistently lower evaluation error compared to Johnson *et al.*, numerically proving the generalizability of the learned representation.

5.3.5 MetaStyle as Preprocessing for Ghiasi *et al.* [GLK17]

As discussed above, we insert MetaStyle into the content transformation branch in the PCIN model proposed by [GLK17]. Clearly, the MetaStyle module could be considered as a preprocessing step. Another possibility in incorporating MetaStyle is treating it as initialization and jointly optimizing it together with other modules. However, in a preliminary study, we notice that treating MetaStyle as fixed preprocessing achieves much lower losses and bet-

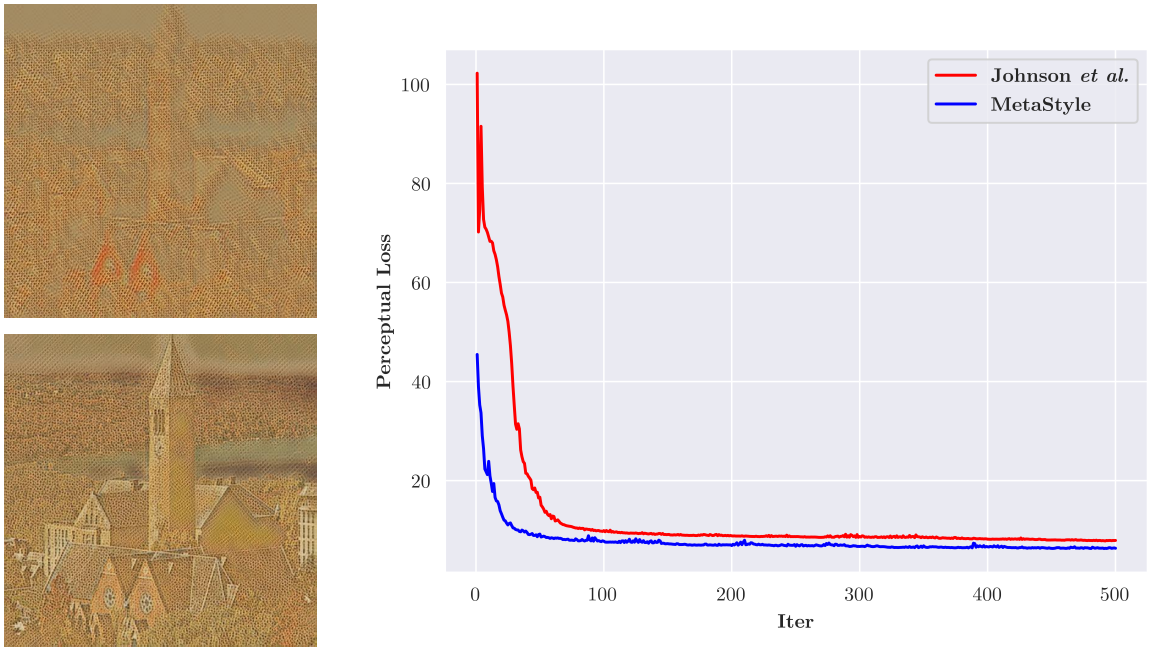

Figure 5.4: Comparison with Johnson *et al.*. (Left) The results using (upper) Johnson *et al.* and (lower) the proposed MetaStyle. (Right) The perceptual loss during evaluation.

ter stylization results. Therefore, here we only report results obtained using MetaStyle as a fixed preprocessing step. Figure 5.5 shows the loss dynamics of PCIN models with and without MetaStyle preprocessing. As shown in the figure, the total loss and the style loss of the MetaStyle version are consistently lower than the original PCIN model. We also show the stylization results in Figure 5.6. While stylization by both methods produces images of satisfactory transfer effects, images stylized by the MetaStyle extention exhibit stronger inheritance from different styles, with more realistic stylized strokes coming from their style sources. This visual effect also coheres with the numerical results, where the style loss obtained from the MetaStyle version is consistently lower. In terms of content loss, although the MetaStyle version has a slightly higher content loss, the semantics in the content sources are extremely well-preserved.

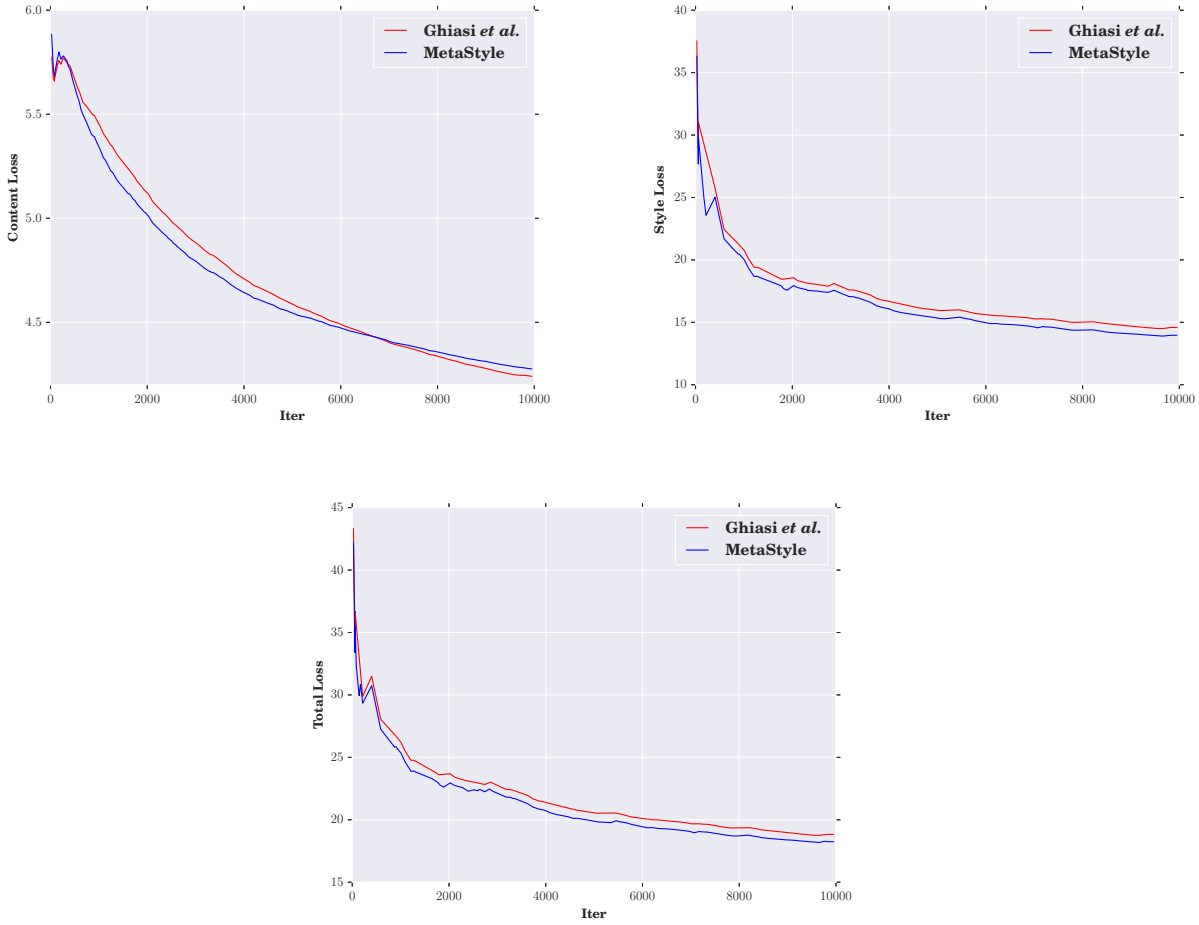

Figure 5.5: Loss dynamics during training of the original PCIN model and the one with MetaStyle preprocessing. As could be seen, although the content loss of the MetaStyle extention is slightly higher, both style loss and the total loss are consistently lower.

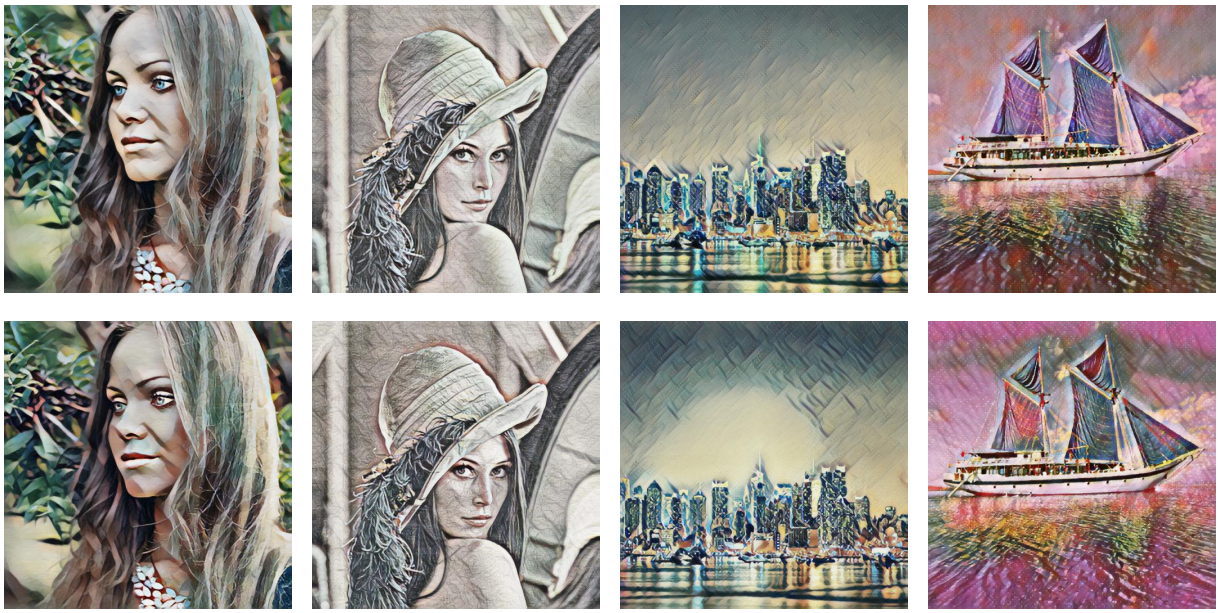

Figure 5.6: Stylization using the original PCIN model and the MetaStyle extention. The first row shows the images stylized by PCIN and the second by MetaStyle PCIN. It could be seen that style inheritance by MetaStyle PCIN is stronger.

CHAPTER 6

Conclusion

In this work, we present the MetaStyle, which is designed to achieve a right three-way trade-off among speed, flexibility, and quality in neural style transfer. Unlike previous methods, MetaStyle considers the arbitrary style transfer problem in a new scenario where a small (even negligible) number of post-processing updates are allowed to adapt the model quickly to a specific style. We formulate the problem in a novel bilevel optimization framework and solve it using MAML. In experiments, we show that MetaStyle could adapt quickly to an arbitrary style within 200 iterations. Each adapted model is an image transformation network and achieves the high efficiency and style transformation quality on par with Johnson *et al.*. We also investigate the representation learned by MetaStyle. We show in experiments that the representation learned by the proposed bilevel optimization could not only generalize in the style interpolation task and video style transfer task, but also work as a plug-in preprocessing step to help the optimization-based method, the fast approximation method, and the complex arbitrary style transfer method converge. Both quantitative and qualitative evaluations show that MetaStyle could serve as a fast, flexibly, and high-quality neural style transfer method, with broadly generalizable style-free representation.

REFERENCES

[CS16] Tian Qi Chen and Mark Schmidt. "Fast patch-based style transfer of arbitrary style." *arXiv preprint arXiv:1612.04337*, 2016.

[CYL17] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. "Stylebank: An explicit representation for neural image style transfer." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[DDS09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[DF81] Persi Diaconis and David Freedman. "On the statistics of vision: the Julesz conjecture." *Journal of Mathematical Psychology*, **24**(2):112–138, 1981.

[DSK17] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. "A learned representation for artistic style." *International Conference on Learning Representations (ICLR)*, 2017.

[EF01] Alexei A Efros and William T Freeman. "Image quilting for texture synthesis and transfer." In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001.

[EL99] Alexei A Efros and Thomas K Leung. "Texture synthesis by non-parametric sampling." In *Proceedings of International Conference on Computer Vision (ICCV)*, 1999.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks." In *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

[FFS18] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. "Bilevel Programming for Hyperparameter Optimization and Meta-Learning." In *Proceedings of International Conference on Machine Learning (ICML)*, 2018.

[GEB15] Leon Gatys, Alexander S Ecker, and Matthias Bethge. "Texture synthesis using convolutional neural networks." In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015.

[GEB16] 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 (CVPR)*, 2016.

[GLK17] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. "Exploring the structure of a real-time, arbitrary neural artistic stylization network." *arXiv preprint arXiv:1705.06830*, 2017.

[HB17] Xun Huang and Serge J Belongie. "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[HJO01] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. "Image analogies." In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001.

[JAF16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*, 2014.

[KEB05] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. "Texture optimization for example-based synthesis." In *ACM Transactions on Graphics (TOG)*, 2005.

[LFY17] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. "Universal style transfer via feature transforms." In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017.

[LLX01] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. "Real-time texture synthesis by patch-based sampling." *ACM Transactions on Graphics (TOG)*, pp. 127–150, 2001.

[LMB14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context." In *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[LW16] Chuan Li and Michael Wand. "Combining markov random fields and convolutional neural networks for image synthesis." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[LWL17] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. "Demystifying neural style transfer." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[NAS18] Alex Nichol, Joshua Achiam, and John Schulman. "On First-Order Meta-Learning Algorithms." *arXiv preprint arXiv:1803.02999*, 2018.

[Nic16] K Nichol. "Painter by numbers, wikiart.", 2016.

[RL16] Sachin Ravi and Hugo Larochelle. "Optimization as a model for few-shot learning." *International Conference on Learning Representations (ICLR)*, 2016.

[SLS18] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. "Avatar-Net: Multi-scale Zero-shot Style Transfer by Feature Decoration." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[SVI16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[SYZ18] Falong Shen, Shuicheng Yan, and Gang Zeng. "Neural Style Transfer via Meta Networks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[SZ14] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556*, 2014.

[ULV16] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. "Texture Networks: Feed-forward Synthesis of Textures and Stylized Images." In *Proceedings of International Conference on Machine Learning (ICML)*, 2016.

[UVL17] Dmitry Ulyanov, Andrea Vedaldi, and Victor S Lempitsky. "Improved Texture Networks: Maximizing Quality and Diversity in Feed-forward Stylization and Texture Synthesis." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[WL00] Li-Yi Wei and Marc Levoy. "Fast texture synthesis using tree-structured vector quantization." In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000.

[ZD17] Hang Zhang and Kristin Dana. "Multi-style generative network for real-time transfer." *arXiv preprint arXiv:1703.06953*, 2017.

[ZLW00] Song-Chun Zhu, Xiu Wen Liu, and Ying Nian Wu. "Exploring Texture Ensembles by Efficient Markov Chain Monte Carlo-Toward a." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 554–569, 2000.

[ZWM98] Song-Chun Zhu, Yingnian Wu, and David Mumford. "Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling." *Proceedings of International Journal of Computer Vision (IJCV)*, pp. 107–126, 1998.

[ZZ11] Mingtian Zhao and Song-Chun Zhu. "Portrait painting using active templates." In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*, 2011.

[ZZZ18] Chi Zhang, Yixin Zhu, and Song-Chun Zhu. "MetaStyle: Three-Way Trade-Off Among Speed, Flexibility, and Quality in Neural Style Transfer." *arXiv preprint arXiv:1812.05233*, 2018.