MetaStyle: Three-Way Trade-Off Among Speed, Flexibility, and Quality in Neural Style Transfer

Chi Zhang and **Yixin Zhu** and **Song-Chun Zhu**
{chizhang,yzhu,sczhu}@cara.ai
International Center for AI and Robot Autonomy (CARA)

Abstract

An unprecedented booming has been witnessed 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 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 with satisfying artistic effects, comparable to the optimization-based methods for an arbitrary style. 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.

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 (Diaconis and Freedman 1981; Zhu, Wu, and Mumford 1998) problem. Some notable work includes: (i) non-parametric sampling methods (Efros and Leung 1999) and acceleration methods by a tree-structured vector quantization (Wei and Levoy 2000), (ii) patch-based sampling methods (Efros and Freeman 2001; Liang et al. 2001) for better quality and efficiency, (iii) energy minimization methods using EM-like algorithms (Kwatra et al. 2005), and (iv) image analogies (Hertzmann et al. 2001) to produce the "filtered" results and their extensions to portrait paintings (Zhao and Zhu 2011).

With the recent boost of deep neural networks and large datasets in computer vision, Gatys, Ecker, and Bethge (2016) first discovered that combining multi-level VGG features (Simonyan and Zisserman 2014) trained on the ImageNet (Deng et al. 2009) 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 (Gatys, Ecker, and Bethge 2015; 2016; Li and Wand 2016) 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 and becoming impractical for deployment in products. Feed-forward neural networks trained with *perceptual loss* (Johnson, Alahi, and Fei-Fei 2016; Dumoulin, Shlens, and Kudlur 2017; Zhang and Dana 2017) 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 (Huang and Belongie 2017; Sheng et al. 2018) 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.

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 (Finn, Abbeel, and Levine 2017; Nichol, Achiam, and Schulman 2018). 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,

Figure 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

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 (Gatys, Ecker, and Bethge 2016), but transferring styles much more effectively.

Our model is instantiated using a neural network. The network structure is inspired by the finding (Dumoulin, Shlens, and Kudlur 2017) that scaling and shifting parameters in instance normalization layers (Ulyanov, Vedaldi, and Lempitsky 2017) 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 Johnson, Alahi, and Fei-Fei (2016), 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 shows sample results using the proposed style transfer.

The contributions of the paper are three-fold:

- 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 paper 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.
- The proposed method results in a style-neutral representation that comes with better convergence for vanilla optimization-based style transfer methods.

2 Related Work

2.1 Neural Style Transfer

By leveraging the pre-trained VGG model (Simonyan and Zisserman 2014), Gatys, Ecker, and Bethge (2016) 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, Johnson, Alahi, and Fei-Fei (2016) recruited an image transformation network to generate stylized results sufficiently close to the optimum solution directly. Concurrent work by Ulyanov et al. (2016) instantiated a similar idea using multi-resolution generator network and further improved the diversity of the generated images (Ulyanov, Vedaldi, and Lempitsky 2017) by applying the Julesz ensembles (Zhu, Wu, and Mumford 1998; Zhu, Liu, and Wu 2000). 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. Li and Wand (2016) modeled the process using an Markov random field (MRF) and introduced the MRF loss for the task. Li et al. (2017a) 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. Chen et al. (2017) jointly learned a style bank for each style during model training. Dumoulin, Shlens, and Kudlur (2017) modified the instance normalization layer (Ulyanov, Vedaldi, and Lempitsky 2017) to condition on each style. Zhang and Dana (2017) 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 (Chen and Schmidt 2016) 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 (Huang and Belongie 2017) 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 Li et al. (2017b) 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 (Sheng et al. 2018) and a meta network that was trained to generate parameters of an image transformation network (Shen, Yan, and Zeng 2018) 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 on par with fast approximation methods, *e.g.*, Johnson, Alahi, and Fei-Fei (2016).

Additionally, our model is also parsimonious, requiring roughly the same number of model parameters as Johnson, Alahi, and Fei-Fei (2016), using merely 0.1 million iterations. In comparisons, *e.g.*, Ghiasi et al. (2017) extended the conditional instance normalization framework (Dumoulin, Shlens, and Kudlur 2017), but required a pre-trained Inception-v3 (Szegedy et al. 2016) 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* (Franceschi et al. 2018) that influences our work. Ravi and Larochelle (2016) first employed an LSTM network as a meta-learner to learn an optimization procedure. Finn, Abbeel, and Levine (2017) 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, Nichol, Achiam, and Schulman (2018) generalized MAML to a family of algorithms and extended it to Reptile. Reptile coupled sequential first-order gradients with advanced optimizers, such as Adam (Kingma and Ba 2014), resulting in an easier implementation, shorter training time and comparable performance. A recent work (Shen, Yan, and Zeng 2018) 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 paper is the first to explicitly cast neural style transfer as the bilevel optimization problem in the initialization strategy branch.

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. Gatys, Ecker, and Bethge (2016) proposed to measure the optimality with a newly defined loss using the trained VGG features, later modified and named as the perceptual loss (Johnson, Alahi, and Fei-Fei 2016). 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

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

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

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

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

The transformation could be efficiently computed by

$$G(x) = \frac{\psi(x)\psi(x)^T}{CHW} \tag{3}$$

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

$$\ell(I_c, I_s, I_x) = \alpha \ell_{\text{content}}(I_c, I_x) + \beta \ell_{\text{style}}(I_s, I_x). \tag{4}$$

3.2 Bilevel Optimization

We formulate the style transfer problem as the bilevel optimization in the form simplified by Franceschi et al. (2018)

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & E(w_\theta, \theta) \\ \text{subject to} \quad & w_\theta = \arg\min_w L_\theta(w), \end{aligned} \tag{5}$$

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.*,

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & E(w_T, \theta) \\ \text{subject to} \quad & w_0 = \Psi(\theta) \\ & w_t = w_{t-1} - \delta \nabla L_\theta(w_{t-1}) \end{aligned} \tag{6}$$

where Ψ initializes w_0, δ is the step size and T the maximum number of steps. Franceschi et al. (2018) proved the convergence of Equation 6 under certain conditions. Though

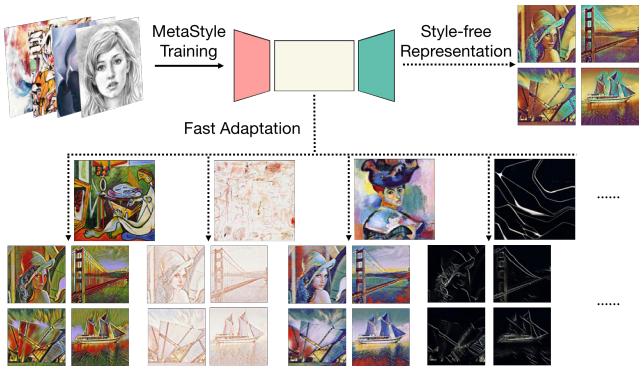

Figure 2: The proposed MetaStyle framework, in which the model is optimized using the bilevel optimization over large-scale content and style dataset. The framework first learns a style-neutral representation. A limited number of post-processing update steps is then applied to adapt the model quickly to a new style. After adaptation, the new model serves as an image transformation network with good transfer quality and high efficiency.

they did not model their problems using bilevel optimization but rather an intuitive motivation, Finn, Abbeel, and Levine (2017) and Nichol, Achiam, and Schulman (2018) 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.

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.

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 non-trivial 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 (Johnson, Alahi, and Fei-Fei 2016) and cast the entire neural style transfer problem in a bilevel optimization framework (Franceschi et al. 2018). As discussed in Equation 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 (Finn, Abbeel, and Levine 2017; Nichol, Achiam, and Schulman 2018), the inner objective uses a model initialized 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

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \mathbb{E}_{c,s}[\ell(I_c, I_s, M(I_c; w_{s,T}))] \\ \text{subject to} \quad & w_{s,0} = \theta \\ & w_{s,t} = w_{s,t-1} - \delta \nabla \mathbb{E}_c[\ell(I_c, I_s, M(I_c; w_{s,t-1}))], \end{aligned} \tag{7}$$

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 2 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.

Relation to Johnson *et al.* (2016): Johnson *et al.*'s method finds an image transformation model tailored to a single given style, minimizing the model parameters by

$$\underset{w}{\text{minimize}} \quad \mathbb{E}_c[\ell(I_c, I_s, M(I_c; w))], \tag{8}$$

where the expectation is taken with respect to only the contents. In contrast, in Equation 7, 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.

Relation to Gatys *et al.* (2016): 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 transformation generates a style-neutral image while dramatically reducing the number of update steps.

Relation to Shen *et al.* (2018): 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.

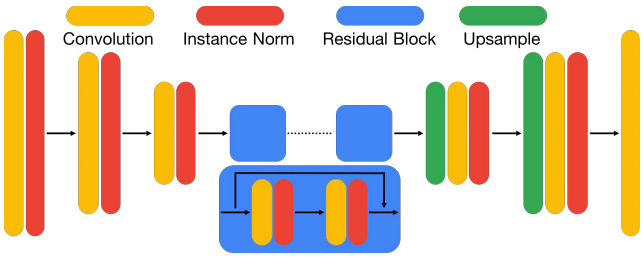

Figure 3: Network architecture. Residual Blocks are stacked multiple times to extract deeper image features.

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 $E \leftarrow 0$
  sample a batch of styles from $\mathcal{D}_{style}$
  for *each style* I_s **do**
    $w_s \leftarrow \theta$
    for $i \leftarrow 1$ **to** T **do**
      sample a batch $\mathcal{B}_{tr}$ from $\mathcal{D}_{tr}$
      compute inner loss L_θ using I_s and $\mathcal{B}_{tr}$
      $w_s \leftarrow w_s - \delta \nabla L_\theta$
    end
    sample a batch $\mathcal{B}_{val}$ from $\mathcal{D}_{val}$
    increment E by loss from I_s and $\mathcal{B}_{val}$
  end
  $\theta \leftarrow \theta - \eta \nabla E$
end

4.2 Network Architecture, Training & Algorithm

Our network architecture largely follows that of an image transformation network described in Dumoulin, Shlens, and Kudlur (2017). 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 (Zhang and Dana 2017). Inspired by the finding (Dumoulin, Shlens, and Kudlur 2017) 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 last. See Figure 3 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.

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 (Finn, Abbeel, and Levine 2017) and summarized in Algorithm 1. After training, θ could be used as the initialization to minimize Equation 8 to adapt the model to a single style or to provide the starting point $M(I_c; \theta)$ for optimization-based methods.

5 Experiments

5.1 Implementation Details

We train our model using MS-COCO (Lin et al. 2014) as our content dataset and WikiArt test set (Nichol 2016) as our style dataset. The content dataset has roughly 80,000 images and the WikiArt test set 20,000 images. We use Adam (Kingma and Ba 2014) 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`. To encourage fast adaptation, we constrain $T = 1$. The entire model is trained on a Nvidia Titan Xp with only 0.1 million iterations.

5.2 Comparison with Existing Methods

We compare the proposed MetaStyle with existing methods (Gatys, Ecker, and Bethge 2016; Johnson, Alahi, and Fei-Fei 2016; Li et al. 2017b; Huang and Belongie 2017; Shen, Yan, and Zeng 2018; Sheng et al. 2018; Chen and Schmidt 2016) 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

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 1: Speed and flexibility benchmarking results. Param lists the number of parameters in each model. 256/512 denotes inputs of $256 \times 256/512 \times 512$. # 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.

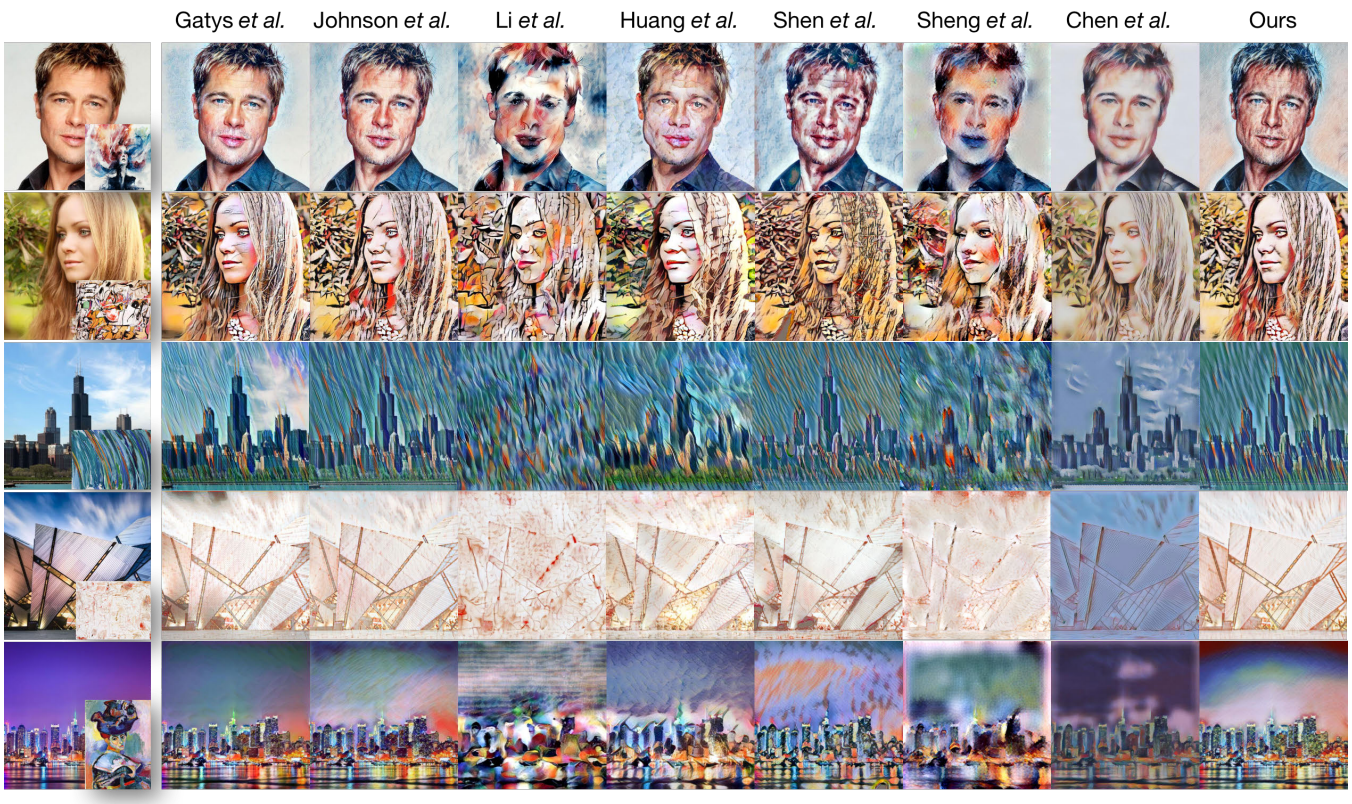

Figure 4: Qualitative comparisons of neural style transfer between the existing methods and the proposed MetaStyle using bilevel optimization. Arbitrary style transfer models observe neither the content images nor the style images during training.

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*..

Speed and Flexibility: Table 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 paragraph.

Quality: Figure 4 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 excessively 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*. (2016), while the fast adaptation comes from our novel formulation; see next paragraph.

Detailed Comparison with Johnson *et al*. (2016): To show the fast adaptation rooted in our formulation, we train a fast approximation model from Johnson *et al*. and adapt our MetaStyle using the same number of updates on a shared style with the same learning rate. Figure 6 shows the results

(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: Style interpolation and video style transfer.

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 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 fast adaptation property of the proposed MetaStyle.

Detailed Comparison with Gatys *et al*. (2016): 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. We empirically illustrate in Figure 7, 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 result. 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 Additional Experiments

We now present two additional experiments to demonstrate the style-neutral representation learned by MetaStyle.

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. Figure 5a shows the results of a two-style interpolation.

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 5b shows the video style transfer results in five consecutive frames. Note that our method does not introduce the flickering effect that harms aesthetics. Additional videos are provided in the supplementary files.

6 Conclusion

In this paper, 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

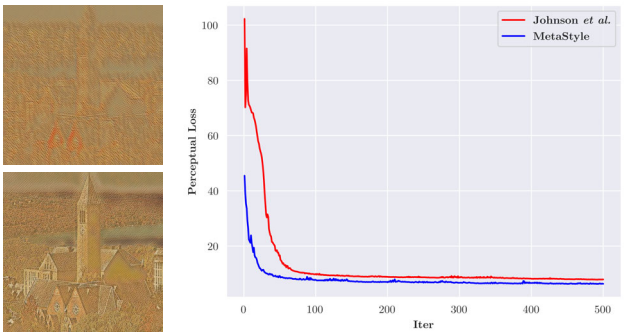

Figure 6: Comparison with Johnson *et al*.. (Left) The results using (upper) Johnson *et al*. and (lower) the proposed MetaStyle. (Right) The perceptual loss during evaluation.

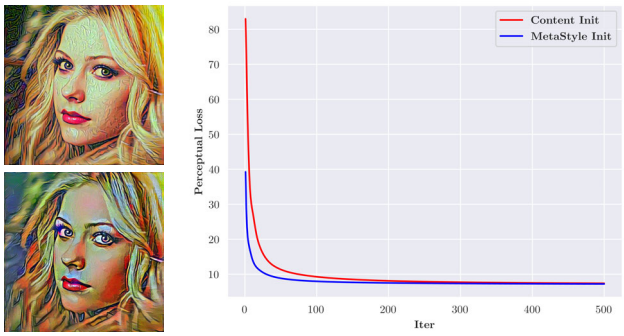

Figure 7: Comparison with Gatys *et al*.. (Left) The results using (upper) Gatys *et al*. and (lower) the proposed MetaStyle. (Right) The perceptual loss.

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 benefits the high efficiency and style transformation quality on par with Johnson *et al*.. The detailed comparison and additional experiments also show the generalized style-neutral representation learned by MetaStyle. These results show MetaStyle indeed achieves a right trade-off.

Acknowledgments: The work reported herein was supported by the International Center for AI and Robot Autonomy (CARA).

References

Chen, T. Q., and Schmidt, M. 2016. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*.

Chen, D.; Yuan, L.; Liao, J.; Yu, N.; and Hua, G. 2017. Stylebank: An explicit representation for neural image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Diaconis, P., and Freedman, D. 1981. On the statistics of vision: the julesz conjecture. *Journal of Mathematical Psychology* 24(2):112–138.

Dumoulin, V.; Shlens, J.; and Kudlur, M. 2017. A learned representation for artistic style. *International Conference on Learning Representations (ICLR)*.

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

Efros, A. A., and Leung, T. K. 1999. Texture synthesis by non-parametric sampling. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of International Conference on Machine Learning (ICML)*.

Franceschi, L.; Frasconi, P.; Salzo, S.; Grazzi, R.; and Pontil, M. 2018. Bilevel programming for hyperparameter optimization and meta-learning. In *Proceedings of International Conference on Machine Learning (ICML)*.

Gatys, L.; Ecker, A. S.; and Bethge, M. 2015. Texture synthesis using convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

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 (CVPR)*.

Ghiasi, G.; Lee, H.; Kudlur, M.; Dumoulin, V.; and Shlens, J. 2017. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830*.

Hertzmann, A.; Jacobs, C. E.; Oliver, N.; Curless, B.; and Salesin, D. H. 2001. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*.

Huang, X., and Belongie, S. J. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of European Conference on Computer Vision (ECCV)*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kwatra, V.; Essa, I.; Bobick, A.; and Kwatra, N. 2005. Texture optimization for example-based synthesis. In *ACM Transactions on Graphics (TOG)*.

Li, C., and Wand, M. 2016. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Y.; Wang, N.; Liu, J.; and Hou, X. 2017a. Demystifying neural style transfer. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017b. Universal style transfer via feature transforms. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

Liang, L.; Liu, C.; Xu, Y.-Q.; Guo, B.; and Shum, H.-Y. 2001. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics (TOG)* 127–150.

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 *Proceedings of European Conference on Computer Vision (ECCV)*.

Nichol, A.; Achiam, J.; and Schulman, J. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Nichol, K. 2016. Painter by numbers, wikiart.

Ravi, S., and Larochelle, H. 2016. Optimization as a model for few-shot learning. *International Conference on Learning Representations (ICLR)*.

Shen, F.; Yan, S.; and Zeng, G. 2018. Neural style transfer via meta networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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 (CVPR)*.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ulyanov, D.; Lebedev, V.; Vedaldi, A.; and Lempitsky, V. S. 2016. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proceedings of International Conference on Machine Learning (ICML)*.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. S. 2017. 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)*.

Wei, L.-Y., and Levoy, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*.

Zhang, H., and Dana, K. 2017. Multi-style generative network for real-time transfer. *arXiv preprint arXiv:1703.06953*.

Zhao, M., and Zhu, S.-C. 2011. Portrait painting using active templates. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*.

Zhu, S.-C.; Liu, X. W.; and Wu, Y. N. 2000. Exploring texture ensembles by efficient markov chain monte carlo-toward a. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 554–569.

Zhu, S.-C.; Wu, Y.; and Mumford, D. 1998. Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *Proceedings of International Journal of Computer Vision (IJCV)* 107–126.

Supplementary Material for MetaStyle: Three-Way Trade-Off Among Speed, Flexibility, and Quality in Neural Style Transfer

Chi Zhang and **Yixin Zhu** and **Song-Chun Zhu**
{chizhang,yzhu,sczhu}@cara.ai
International Center for AI and Robot Autonomy (CARA)

1 Details of the Network Architecture

We provide further details on the network architecture used in MetaStyle in Table 1. 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.

2 Additional Details of the Training

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

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

where κ denotes the learning rate for either the outer or the inner objective, t the number of iterations, 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 Finn, Abbeel, and Levine (2017) and Nichol, Achiam, and Schulman (2018), 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 significantly reduce 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 Dumoulin, Shlens, and Kudlur (2017).

3 Additional Neural Style Transfer Results

We include more examples in Page 3-8.

References

Dumoulin, V.; Shlens, J.; and Kudlur, M. 2017. A learned representation for artistic style. *International Conference on Learning Representations (ICLR)*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of International Conference on Machine Learning (ICML)*.

Nichol, A.; Achiam, J.; and Schulman, J. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

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 1: Network architecture used in MetaStyle.