A Closed-Form Solution to Universal Style Transfer

Ming Lu *[1], Hao Zhao[1], Anbang Yao[2], Yurong Chen[2], Feng Xu[3], and Li Zhang[1]

[1]Department of Electronic Engineering, Tsinghua University
[2]Intel Labs China
[3]BNRist and School of Software, Tsinghua University
{lu-m13@mails,zhao-h13@mails,feng-xu@mail,chinazhangli@mail}.tsinghua.edu.cn
{anbang.yao, yurong.chen}@intel.com

Abstract

Universal style transfer tries to explicitly minimize the losses in feature space, thus it does not require training on any pre-defined styles. It usually uses different layers of VGG network as the encoders and trains several decoders to invert the features into images. Therefore, the effect of style transfer is achieved by feature transform. Although plenty of methods have been proposed, a theoretical analysis of feature transform is still missing. In this paper, we first propose a novel interpretation by treating it as the **optimal transport** *problem. Then, we demonstrate the relations of our formulation with former works like Adaptive Instance Normalization (AdaIN) and Whitening and Coloring Transform (WCT). Finally, we derive a closed-form solution named* **Optimal Style Transfer** *(OST) under our formulation by additionally considering the content loss of Gatys. Comparatively, our solution can preserve better structure and achieve visually pleasing results. It is simple yet effective and we demonstrate its advantages both quantitatively and qualitatively. Besides, we hope our theoretical analysis can inspire future works in neural style transfer. Code is available at* `https://github.com/lu-m13/OptimalStyleTransfer`.

1. Introduction

A variety of methods on neural style transfer have been proposed since the seminal work of Gatys [8]. These methods can be roughly categorized into image optimization and model optimization [13]. Methods based on image optimization directly obtain the stylized output by minimizing the content loss and style loss. The style loss can be defined by Gram matrix [8], histogram [25], or Markov Random Fields (MRFs) [16]. Contrary to that, methods based on model optimization try to train neural networks on large datasets like COCO [22]. The training loss can be defined as perceptual loss [14] or MRFs loss [17]. Subsequent works [3, 6, 32] further study the problem of training one network for multiple styles. Recently, [12] proposes to use AdaIN as feature transform to train one network for arbitrary styles. Apart from image and model optimization, many other works study the problems of semantic style transfer [23, 21, 1], video style transfer [11, 2, 26, 27], portrait style transfer [28], and stereoscopic style transfer [4]. [13] provides a thorough review of the works on style transfer.

In this paper, we study the problem of universal style transfer [19]. Our motivation is to explicitly minimize the losses defined by Gatys [8]. Therefore, our approach does not require training on any pre-defined styles. Similar to WCT [19], our method is also based on a multi-scale encoder-feature transform-decoder framework. We use different layers of VGG network [31] as the encoders and train the decoders to invert features into images. The effect of style transfer is achieved by feature transform between encoder and decoder. Therefore, the key to universal style transfer is feature transform. In this work, we focus on the theoretical analysis of feature transform and propose a new closed-form solution.

Although AdaIN [12] trains its decoder on a large dataset of style images, AdaIN itself is also a feature transform method. It considers the feature of each channel as a Gaussian distribution and assumes the channels are independent. For each channel, AdaIN first normalizes the content feature and then matches it to the style feature. This means it only matches the diagonal elements of the covariance matrices. WCT [19] proposes to use whitening and coloring as feature transform. Compared with AdaIN, WCT improves the

*This work was done when Ming Lu was an intern at Intel Labs China, supervised by Anbang Yao who is responsible for correspondence.

results by matching all the elements of covariance matrices. Since the channels of deep Convolutional Neural Networks (CNNs) are correlated, the non-diagonal elements are essential to represent the style. However, WCT only matches the covariance matrices, which shares similar spirits with minimizing the style loss of Gatys. It does not consider the content loss and cannot well preserve the image structure. Moreover, multiplying an orthogonal matrix between the whitening and coloring matrices can also match the covariance matrices, which has been pointed out by [18].

[20] shows that matching Gram matrices is equivalent to minimizing the Maximum Mean Discrepancy (MMD) with the second order polynomial kernel. However, it does not give a closed-form solution. Instead, our work reformulates style transfer as an optimal transport problem. Optimal transport tries to find a transformation that matches two high-dimensional distributions. For neural style transfer, considering the neural feature in each activation as a high dimension sample, we assume the samples of content and style images are from two Multivariate Gaussian (MVG) distributions. Style transfer is equivalent to transforming the content samples to fit the distribution of style samples. Assuming the transformation is linear, we find that both AdaIN and WCT are special cases of our formulation. Although [18] also assumes the transformation is linear, it still follows the whitening and coloring pipeline and trains two meta networks for whitening and coloring matrices. Contrary to that, we directly find the transformation under the optimal transport formulation.

As we have described above, there are still infinite transformations, for example, multiplying an orthogonal matrix between the whitening and coloring matrices can also be the solution. Therefore, we seek for a transformation, which additionally minimizes the difference between transformed feature and original content feature. This shares similar spirits with minimizing the content loss of Gatys [8]. We prove that a unique closed-form solution named Optimal Style Transfer (OST) can be found, once considering the content loss. We show the detailed proof of OST in the method part. Since OST further considers the content loss, it can preserve better structures compared with WCT.

Our contributions can be concluded as follows:

1. We present a novel interpretation of neural style transfer by treating it as an optimal transport problem and elucidate the theoretical relations of our interpretation with former works on feature transform, for example, AdaIN and WCT.

2. We find the unique closed-form solution named OST under the optimal transport interpretation by additionally considering the content loss.

3. Our closed-form solution preserves better structures and achieves visually pleasing results.

2. Related Work

Image Optimization. Methods based on image optimization directly obtain the stylized output by minimizing the content loss and style loss defined in the feature space. The optimization is usually based on back-propagation. [7, 8] propose to use Gram matrix to define the style of an example image. [16] improves the results by combining MRFs with Convolutional Neural Networks. [1] uses the semantic masks to define the style losses within corresponding regions. In order to improve the results for portrait style transfer, [28] proposes to modify the feature maps to transfer the local color distributions of the example painting onto the content image. This is similar to the gain map proposed by [30]. [9] studies the problem of controlling the perceptual factors during style transfer. [25] improves the results of neural style transfer by incorporating the histogram loss. [26] incorporates the temporal consistency loss into the optimization for video style transfer. Since all the above methods solve the optimization by back-propagation, they are intrinsically time-consuming.

Model Optimization. In order to solve the speed bottleneck of back-propagation, [14, 17] propose to train a feed-forward network to approximate the optimization process. Instead of optimizing the image, they optimize the parameters of the network. Since it is tedious to train one network for each style, [3, 6, 32] further study the problem of training one network for multiple styles. Later, [5] presents a method based on patch swap for arbitrary style transfer. First, the content and style images are forwarded through the deep neural network to extract features. Then the style transfer is formulated as neural patch swap to get the reconstructed feature map. This feature map is inverted by the decoder network to image space. Since then, the framework of encoder-feature transform-decoder has been widely explored for arbitrary style transfer. [12] uses AdaIN as the feature transform and trains the decoder over large collections of content and style images. [18] trains two meta networks for the whitening and coloring matrices, following the formulation of WCT [19]. Many other works also extend neural style transfer to video [2, 11, 27] and stereoscopic style transfer [4]. These works usually jointly train additional networks apart from the style transfer network.

Universal Style Transfer. Universal style transfer [19] is also based on the framework of encoder-feature transform-decoder. Unlike AdaIN [12], it does not require network training on any style image. It directly uses different layers of VGG network as the encoders and train the decoders to invert the feature into image. The style transfer effect is achieved by feature transform. [5] replaces the patches of content feature by the most similar patches of style feature. However, the nearest neighbor search achieves less transfer effect since it tends to preserve the original appearance. AdaIN considers the activa-

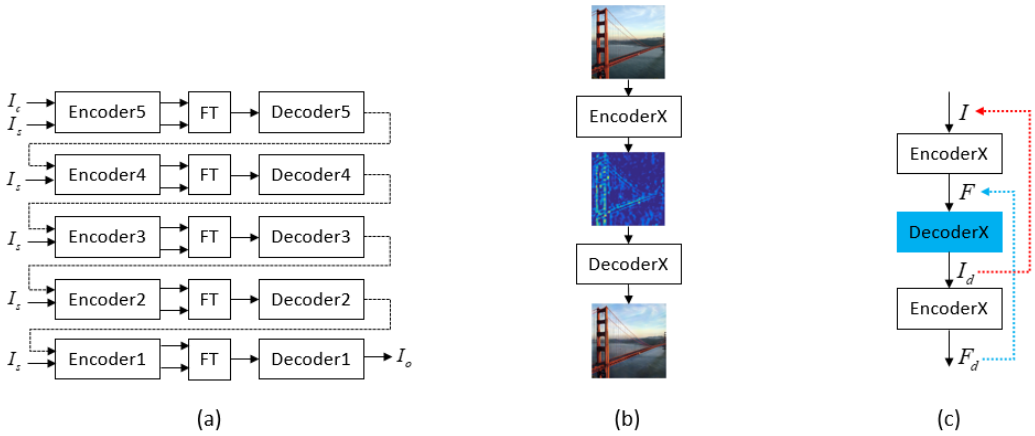

Figure 1. (a) The pipeline of OST for universal style transfer. First, we extract features using the encoder for content image and style image. Then we use the feature transform method to obtain the stylized feature. Finally, the decoder inverts the stylized feature into image. The output of top layer is used as the input content image for the bottom layer. (b) The decoder inverts the feature of a certain layer to the image. Although [10, 29] propose to train the decoder to invert the feature to its bottom layer's feature, which might be more efficient, we use the image decoder in this work since decoder is not our contribution. (c) We use the feature loss (denoted by the blue arrow) and the reconstruction loss (denoted by the red arrow) to train the DecoderX (X=1,2,...,5).

tion of each channel as a Gaussian distribution and matches the content and style images through mean and variance. However, since the channels of CNN are correlated, AdaIN cannot achieve visually pleasing transfer effect. WCT [19] proposes to use feature whitening and coloring to match the covariance matrices of style and content images. However, as pointed out by [18], WCT is not the only approach to matching the covariance matrices. [29] proposes a method to combine patch match with WCT and AdaIN. Instead of finding the nearest neighbor by the original feature, [29] conducts it using the projected feature. These projected feature can be generated by AdaIN or WCT. However, above methods all fail to give a theoretical analysis of feature transform. The key observation of current works like WCT is matching the covariance matrices, which is not enough to find a good solution.

3. Motivation

The pipeline of OST is shown in Figure 1. It is similar to WCT [19]. We use different layers of the pre-trained VGG network as the encoders. For every encoder, we train the corresponding decoder to invert the feature into image as illustrated by Figure 1 (b, c). Although [10, 29] propose to train the decoder to invert the feature to its bottom layer's feature, which might be more efficient, we use the image decoder [19] in this work since the framework is not our contribution.

We start to study the problem of feature transform by reformulating neural style transfer as the optimal transport problem. We denote the content image as I_c and the style image as I_s. For the features of content image and style image, we denote them as $F_c \in R^{C\times H_cW_c}$ and $F_s \in R^{C\times H_sW_s}$ separately, where H_cW_c and H_sW_s are the numbers of activations and C is the number of channels. We view the columns of F_c and F_s as samples from two Multivariate Gaussian (MVG) distributions $N(\mu_c, \Sigma_c)$ and $N(\mu_s, \Sigma_s)$, where $\mu_c, \mu_s \in R^C$ are the mean vectors and $\Sigma_c, \Sigma_s \in R^{C\times C}$ are the variance matrices. We further denote the sample from content distribution as u and the sample from style distribution as v. Therefore, $u \sim N(\mu_c, \Sigma_c)$ and $v \sim N(\mu_s, \Sigma_s)$. Assuming the optimal transformation is linear, we can represent it as follows.

$$t(u) = T(u - \mu_c) + \mu_s \tag{1}$$

Where $T \in R^{C\times C}$ is the transformation matrix. Since we assume the features are from two MVG distributions, T must meet the following equation to match two MVG distributions.

$$T\Sigma_c T^T = \Sigma_s \tag{2}$$

Where T^T is the transpose of T. When Eq. 2 is satisfied, we can obtain $t(u) \sim N(\mu_s, \Sigma_s)$. We then demonstrate the relations of our formulation with AdaIN [12] and WCT [19]. We denote the diagonal matrices of Σ_c and Σ_s as D_c and D_s separately. **For AdaIn**, the transformation matrix $T = D_s./D_c$, where $./$ denotes the element-wise division. Therefore, AdaIN does not satisfy Eq. 2 since it ignores the correlation of channels. Only the diagonal elements are matched by AdaIN. **As for WCT**, we can find that the transformation matrix $T = \Sigma_s^{1/2}\Sigma_c^{-1/2}$. Since both $\Sigma_s^{1/2}$ and $\Sigma_c^{-1/2}$ are symmetric matrices, WCT satisfies Eq.

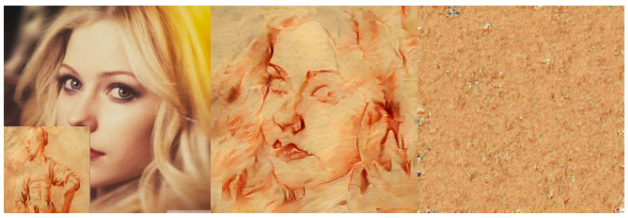

Figure 2. Style transfer results of $T = \Sigma_s^{1/2} Q \Sigma_c^{-1/2}$, where Q is a unite orthogonal matrix. Although T satisfies Eq. 2, the results vary significantly.

2. However, WCT is not the only solution to Eq. 2 because $T = \Sigma_s^{1/2} Q \Sigma_c^{-1/2}$, where Q is a unite orthogonal matrix, is a family of solutions to Eq. 2. This has also been pointed out by [18]. Theoretically, there are infinite solutions, considering only Eq. 2. We show the style transfer results of multiplying a random unite orthogonal matrix to the whitening matrix in Figure 2. As can been seen, although $T = \Sigma_s^{1/2} Q \Sigma_c^{-1/2}$ satisfies Eq. 2, the style transfer results vary significantly.

Our motivation is to find an optimal solution by additionally considering the content loss of Gatys. Therefore, our formulation can be represented as follows, where E represents the expectation.

$$\begin{aligned} T &= \arg\min_T E(||t(u) - u||_2^2) \\ s.t. \quad t(u) &= T(u - \mu_c) + \mu_s \\ T\Sigma_c T^T &= \Sigma_s \end{aligned} \tag{3}$$

4. Method

In this part, we derive the closed-form solution to Eq. 3. We substitute Eq. 1 to the expectation term of Eq. 3 and obtain:

$$E[(T(u - \mu_c) + \mu_s - u)^T (T(u - \mu_c) + \mu_s - u)] \tag{4}$$

We denote $u^* = u - \mu_c$, $v^* = Tu^*$ and $\delta = \mu_s - \mu_c$. Therefore, we can get $u^* \sim N(0, \Sigma_c)$ and $v^* \sim N(0, \Sigma_s)$. Besides, δ is a constant C-dimensional vector. Using u^*, v^* and δ, we can re-write Eq. 4 as:

$$E[(v^* + \delta - u^*)^T (v^* + \delta - u^*)] \tag{5}$$

We further expand Eq. 5 to:

$$\begin{aligned} E[&v^{*T} v^* + \delta^T v^* - u^{*T} v^* + v^{*T} \delta + \delta^T \delta \\ &- u^{*T} \delta - v^{*T} u^* - \delta^T u^* + u^{*T} u^*] \end{aligned} \tag{6}$$

Since $u^* \sim N(0, \Sigma_c)$, $v^* \sim N(0, \Sigma_s)$ and δ is a constant C-dimensional vector, we can get $E[\delta^T v^*] = E[v^{*T} \delta] = 0$ and $E[u^{*T} \delta] = E[\delta^T u^*] = 0$. Besides, $E[\delta^T \delta]$ is also constant. Therefore, minimizing Eq. 6 is equivalent to minimizing Eq. 7:

$$E[v^{*T} v^* + u^{*T} u^* - u^{*T} v^* - v^{*T} u^*] \tag{7}$$

Using the representation of matrix trace, Eq. 7 can be rewritten as follows.

$$tr(E[v^* v^{*T} + u^* u^{*T} - v^* u^{*T} - u^* v^{*T}]) \tag{8}$$

Where tr means the trace of a matrix. Since $E[v^* v^{*T}] = \Sigma_s$, $E[u^* u^{*T}] = \Sigma_c$ and $E[v^* u^{*T}] = E[u^* v^{*T}] = \phi$, where ϕ denotes the covariance matrix of v^* and u^*, the solution to Eq. 3 can be reformulated as follows.

$$T = \arg\max_T (tr(\phi)) \tag{9}$$

Next, we introduce a lemma, which has been proved by [24]. We do not repeat the proof due to limited space. The lemma can be concluded as follows.

Lemma 4.1 *Given two high-dimensional distributions X and Y, where $X \sim N(0, \Sigma_{11})$ and $Y \sim N(0, \Sigma_{22})$, we define the distribution of (X, Y) as $N(0, \Sigma)$, where Σ can be represented as follows.*

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \phi \\ \phi^T & \Sigma_{22} \end{pmatrix} \tag{10}$$

The problem of $\max(tr(2\phi))$ has a unique solution, which can be represented as:

$$\phi = \Sigma_{11} \Sigma_{22}^{1/2} (\Sigma_{22}^{1/2} \Sigma_{11} \Sigma_{22}^{1/2})^{-1/2} \Sigma_{22}^{1/2} \tag{11}$$

With the above lemma, let $X = v^*$, $Y = u^*$, $\Sigma_{11} = \Sigma_s$ and $\Sigma_{22} = \Sigma_c$, we can obtain the solution to Eq. 9, which can be represented as $\phi = \Sigma_s \Sigma_c^{1/2} (\Sigma_c^{1/2} \Sigma_s \Sigma_c^{1/2})^{-1/2} \Sigma_c^{1/2}$. We rewrite the covariance matrix as $\phi = E[v^* u^{*T}] = E[v^* (T^{-1} v^*)^T] = E[v^* v^{*T}](T^{-1})^T = \Sigma_s (T^{-1})^T$. Therefore, we can get $(T^{-1})^T = \Sigma_c^{1/2} (\Sigma_c^{1/2} \Sigma_s \Sigma_c^{1/2})^{-1/2} \Sigma_c^{1/2}$. Then the final T can be represented as Eq. 12.

$$T = \Sigma_c^{-1/2} (\Sigma_c^{1/2} \Sigma_s \Sigma_c^{1/2})^{1/2} \Sigma_c^{-1/2} \tag{12}$$

Remarks: The final solution of our method is very simple. Since our method additionally considers the content loss, we can preserve better structure compared with WCT. Contrary to former works, we provide a complete theoretical proof of the proposed method. The relations of our method with former works are also demonstrated. We believe both the closed-form solution and the theoretical proof will inspire future works in neural style transfer.

Gatys	Patch Swap	AdaIn	AdaIn+	WCT	Ours
207.12s	13.15s	0.49s	0.16s	3.47s	4.06s

Table 1. Processing speed comparison.

5. Results

In this section, we first qualitatively compare our method with Gatys [8], Patch Swap [5], AdaIN (with our decoder) [12], AdaIN+ (with their decoder) [12], and WCT [19] in Section 5.1. Then we provide a quantitative comparison of our method against Gatys, Patch Swap, AdaIN, AdaIN+ and WCT in Section 5.2. Following former works, we also show results of linear interpolation and semantic style transfer in Section 5.3. Finally, we discuss the limitations of our method in Section 5.4.

Parameters: We train the decoders on the COCO dataset [22]. The weight to balance the feature loss and reconstruction loss in Eq. 13 is set to 1 as [19]. For the results in this work, the resolution of the input is fixed as 512×512.

Performance: We implement the proposed method on a server with an NVIDIA Titan Xp graphics card. The processing speed comparison is listed in Table 1 under the input resolution of 512×512. We do the comparison with the published implementations on our server, which might result in slight differences with the papers.

5.1. Qualitative Results

Our Method versus Gatys: Gatys [8] is the pioneering work of neural style transfer and it can handle arbitrary styles. Although it uses time-consuming back-propagation to minimize the content loss and style loss, we still compare with it since its formulation is the foundation of our method. As shown in Figure 3, Gatys can usually achieve reasonable results, however, these results are not so stylized since the iterative solver cannot reach the optimal solution in limited iterations. Instead, our method tries to find the closed-form solution, which explicitly minimizes the style loss and content loss. Comparatively, our results are more stylized and they also well-preserve the structures of content images.

Our Method versus Patch Swap: As far as we know, Patch Swap [5] is the first work to use the encoder-feature transform-decoder framework. It chooses a certain layer of VGG network as the encoder and trains the corresponding decoder. The feature transform is formulated as neural patch swap. However, neural patch swap using the original feature tends to simply reconstruct the feature, thus the results are not stylized. Besides, Patch Swap only transfers the style in a certain layer, which also reduces the style transfer effect. [29] proposes to match the neural patch in the projected domains, for example, the whitened feature [19]. Apart from this, [29] uses multiple layers to transfer the style, achieving more stylized results. Our work does not use the idea of neural patch match, instead, we focus on the theoretical analysis to deliver the closed-form solution. As can be seen in Figure 3, our result is more stylized compared with Patch Swap.

Our Method versus AdaIN and AdaIN+: As discussed in the motivation, AdaIN [12] assumes the channels of CNN feature are independent. For each channel, AdaIN matches two one-dimensional Gaussian distributions. However, the channels of CNN feature are actually correlated. Therefore, using AdaIN as the feature transform cannot achieve visually stylized results. Instead of using AdaIN as the feature transform method, AdaIN+ [12] trains a decoder on large collections of content and style images. Although AdaIN+ only transfers the feature in a certain layer, it trains the decoder with style losses defined in multiple layers. We conduct the comparisons with both AdaIN and AdaIN+. As illustrated by Figure 3, the results of AdaIN and AdaIN+ are similar and both of them fail to achieve visually pleasing transfer results. Therefore, we believe the reason why AdaIN and AdaIN+ fail is because they ignore the correlation between channels of CNN feature. Instead, our work considers the correlation thus achieves more stylized results as shown in Figure 3.

Our Method versus WCT: WCT [19] proposes to use feature whitening and coloring as the solution to style transfer. It chooses ZCA whitening in the paper and we test some other whitening methods with the feature of ReLU3_1 as shown in Figure 4. As can be seen, only ZCA whitening achieves reasonable results. This is because ZCA whitening is the optimal choice, which minimizes the difference between content feature and the whitened feature. Although the ZCA-whitened image can preserve the structure of content image, there is none constraint on the final transformed feature. Contrary to that, we consider to minimize the difference between content feature and the final transformed feature. As we have analyzed in the motivation section, WCT satisfies Eq. 2. Therefore, it perfectly matches two high-dimension Gaussian distributions. However, it ignores the content loss of Gatys. Instead, we seek the closed-form solution, which additionally minimizes the content loss. As can be seen in Figure 3, our transformation can preserve better structures (see the red rectangles).

We also notice that the final feature can be the linear combination of original content feature and the transformed feature as shown in Eq. 13. Where α is the weight of transformed feature.

$$t^*(u) = \alpha t(u) + (1 - \alpha)u \tag{13}$$

We show the results of different α values in Figure 5. As illustrated by Figure 5, adjusting the weight can change the degree of style transfer. With smaller α, WCT can preserve more structure of content image. However, there is still obvious artifact even with small α. Instead, our method

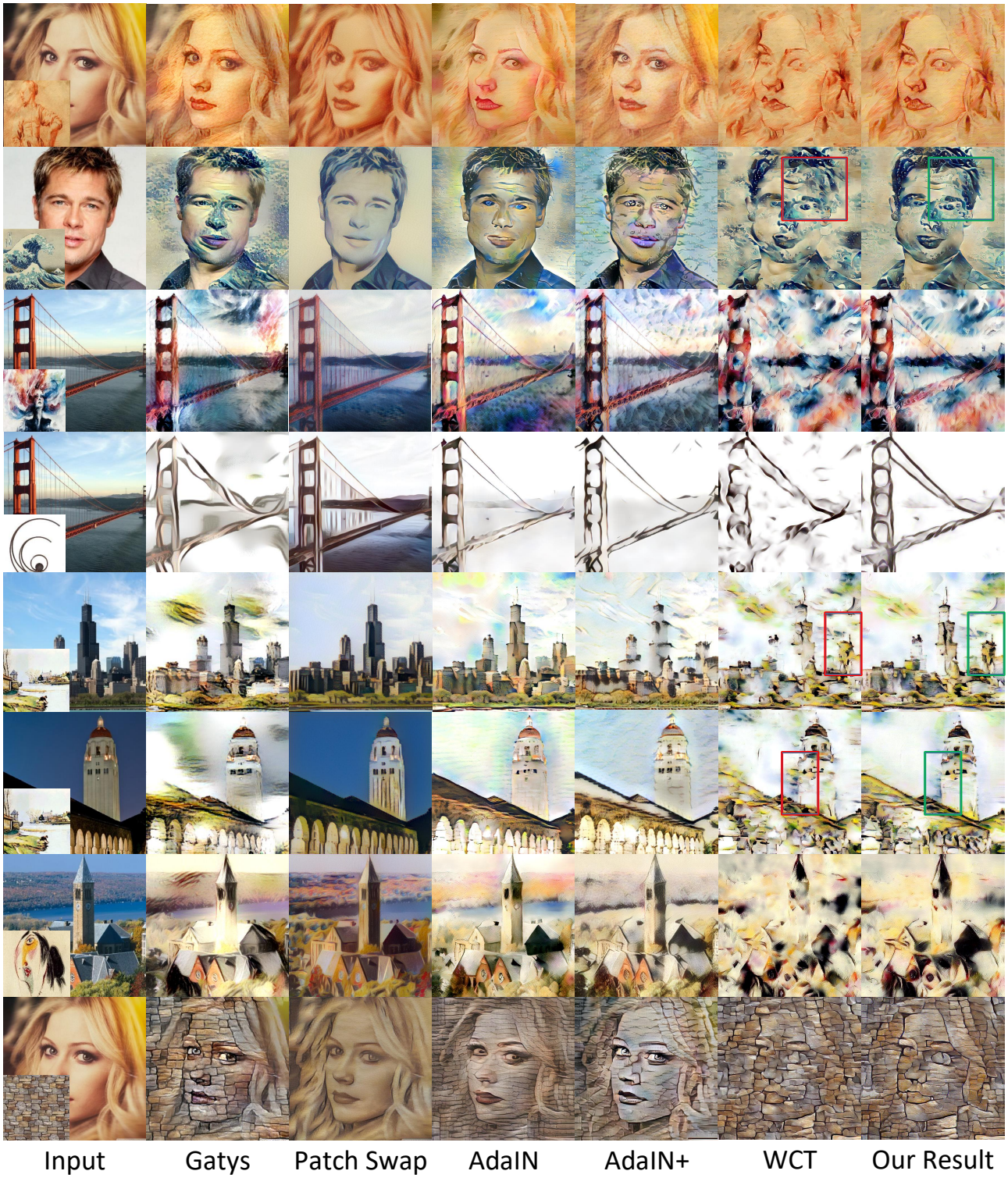

Figure 3. Qualitative results. We compare our method against Gatys [8], Patch Swap [5], AdaIN (with our decoder) [12], AdaIN+ (with their decoder) [12], and WCT [19]. AdaIN ignores the non-diagonal elements of covariance matrices, which results in less stylized output. WCT does not consider the content loss and cannot well-preserve the structure of content image as shown in the red rectangles. Our method can achieve both stylized and content-preserving results.

Method	Gatys	Patch Swap	AdaIN	AdaIN+	WCT*	Ours*
Content Loss	0.096	0.086	0.167	0.151	**0.296**	**0.255**
Style Loss-1	23.77	100.8	15.85	15.9	3.89	3.60
Style Loss-2	8577.04	30647.6	5351.6	3355.5	594.4	457.8
Style Loss-3	6749.7	15607.2	4564.7	4905.5	1226.6	1203
Style Loss-4	325939	562192	245133	202767	187907	129695
Style Loss-5	15.96	17.73	14.1	12.48	24.33	12.37

Table 2. Average content loss and style losses. * means fully matching the statistics of content and style features.

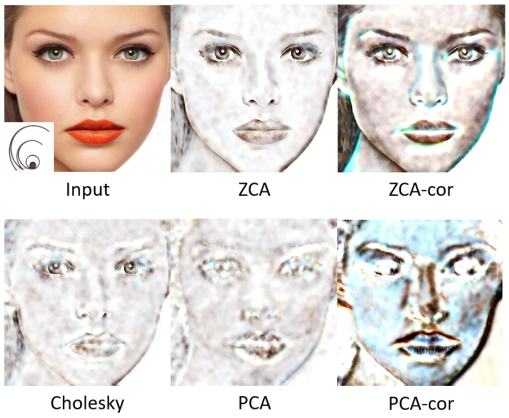

Figure 4. Illustration of different whitening methods. We test some whitening methods with the feature of ReLU3_1. As can be seen, ZCA whitening achieves better results. "cor" means correlated and details of whitening methods can be found in [15]

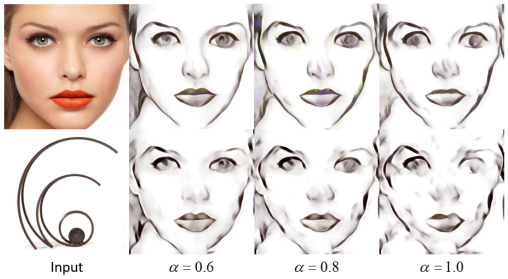

Figure 5. Illustration of linear interpolation. The top row is the results of our method and the bottom row is the results of WCT. Linearly combining content feature with the transformed feature can help preserve the structure. With smaller α in Eq. 13, WCT can preserve more structure. However, there is still obvious artifact. Instead, our method consistently achieves pleasing results.

Gatys	Patch Swap	AdaIN	AdaIN+	WCT	Ours
2.17	1.05	2.00	1.94	2.67	3.07

Table 3. Average scores of user study.

consistently achieves visually pleasing results.

5.2. Quantitative Results

User Study: Style transfer is a very subjective research topic. Although we have theoretically proved the advantages of our method, we further conduct a user study to quantitatively compare our work against Gatys, Patch Swap, AdaIN, AdaIN+ and WCT. This study uses 16 content images and 35 style images collected from published implementations, thus 560 stylized images are generated by each method. We show the content, style and stylized images to testers. We ask the testers to choose a score from 1 (worst) - 5 (best) for the purpose of evaluating the quality of style transfer. We do this user study with 50 testers online. The average scores are listed in the Table 3. This study shows that our method improves the results of former works.

Content Loss and Style Loss: In addition to user study, we also evaluate the content loss and style loss defined by Gatys [8]. We calculate the average content loss and style loss with the images of user study for each method. We normalize the content loss with the number of neural activations. The average losses are listed in Table 2. As can be seen, compared with WCT, our method achieves lower content loss and similar style loss. As for Gatys, Patch Swap, AdaIN and AdaIN+, they fail to achieve stylized results with high style losses as we have analyzed in the qualitative comparison part.

5.3. More Results

We show more results to demonstrate the generalization of our method in Figure 6, where α is set as 1. To further evaluate the linear interpolation, we show two samples with different α values in Figure 7. We also combine our method with semantic style transfer as shown in Figure 7. Although we assume the neural features are sampled from MVG distributions in the proof, these results are all visually pleasing, which demonstrate the generalization ability of the proposed method.

5.4. Limitations

Our method still has some limitations. For example, we evaluate the frame-by-frame results of video style transfer. Although our method can preserve better structure compared with former works, the frame-by-frame results still contain obvious jittering. We find that the temporal jittering is not only caused by feature transform but also caused by the information loss of encoder networks. Deep encoder network will cause obvious temporal jittering even without feature transform.

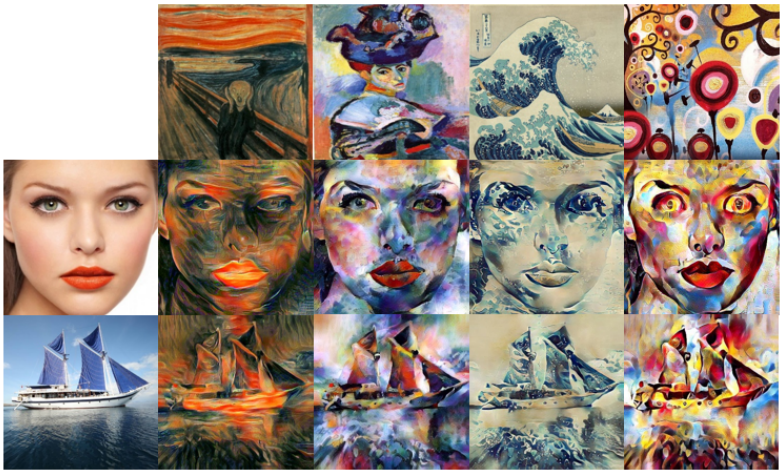

Figure 6. More results. We show more results, where α is set as 1.

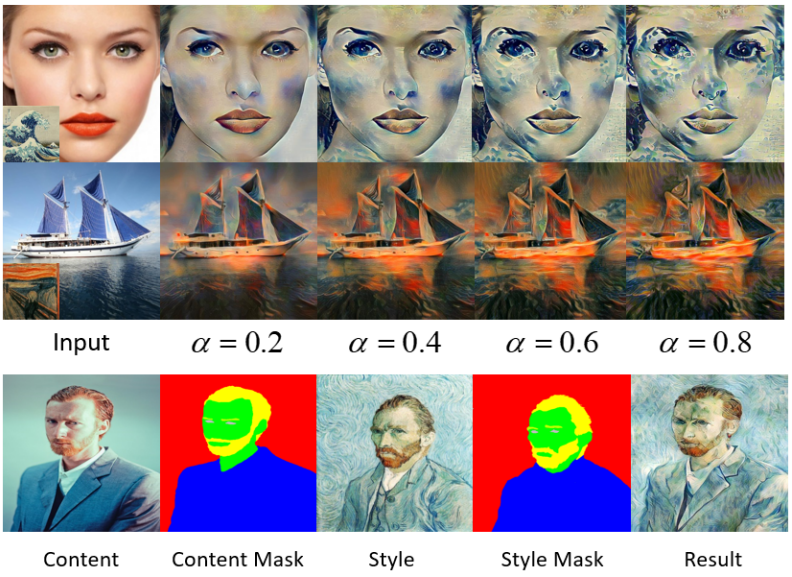

Figure 7. Linear interpolation and semantic style transfer. Although we assume the neural features are sampled from MVG distributions in the proof, these results are all visually pleasing, which demonstrate the generalization ability of our work.

Besides, style transfer is a very subjective problem. Although the Gram matrix representation proposed by Gatys has been widely used, mathematically modeling of what people really feel about style is still an unsolved problem. Exploring the relation between deep neural network and image style is an interesting topic.

6. Conclusion

In this paper, we first present a novel interpretation of neural style transfer by treating it as an optimal transport problem. Then we demonstrate the theoretical relations between our interpretation and former works, for example, AdaIN and WCT. Based on our formulation, we derive the unique closed-form solution by additionally considering the content loss. Our solution preserves better structure compared with former works due to the minimization of content loss. We hope this paper can inspire future works in style transfer.

Acknowledgements. This work was supported by the National Key R&D Program of China 2018YFA0704000, the NSFC (Grant No. 61822111, 61727808, 61671268, 61132007, 61172125, 61601021, and U1533132) and Beijing Natural Science Foundation (L182052).

References

[1] Alex J Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv preprint arXiv:1603.01768*, 2016.

[2] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent online video style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1105–1114, 2017.

[3] 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*, pages 1897–1906, 2017.

[4] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stereoscopic neural style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6654–6663, 2018.

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

[6] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style.

[7] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in neural information processing systems*, pages 262–270, 2015.

[8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[9] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3985–3993, 2017.

[10] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. Arbitrary style transfer with deep feature reshuffle. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8222–8231, 2018.

[11] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. Real-time neural style transfer for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 783–791, 2017.

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

[13] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *arXiv preprint arXiv:1705.04058*, 2017.

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[15] Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.

[16] 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*, pages 2479–2486, 2016.

[17] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[18] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast arbitrary style transfer. *arXiv preprint arXiv:1808.04537*, 2018.

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

[20] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017.

[21] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017.

[22] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[23] Ming Lu, Hao Zhao, Anbang Yao, Feng Xu, Yurong Chen, and Li Zhang. Decoder network over lightweight reconstructed feature for fast semantic style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2469–2477, 2017.

[24] Ingram Olkin and Friedrich Pukelsheim. The distance between two random vectors with given dispersion matrices. *Linear Algebra and its Applications*, 48:257–263, 1982.

[25] Eric Risser, Pierre Wilmot, and Connelly Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint arXiv:1701.08893*, 2017.

[26] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *German Conference on Pattern Recognition*, pages 26–36. Springer, 2016.

[27] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos and spherical images. *International Journal of Computer Vision*, 126(11):1199–1219, 2018.

[28] Ahmed Selim, Mohamed Elgharib, and Linda Doyle. Painting style transfer for head portraits using convolutional neural networks. *ACM Transactions on Graphics (ToG)*, 35(4):129, 2016.

[29] 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*, pages 8242–8250, 2018.

[30] YiChang Shih, Sylvain Paris, Connelly Barnes, William T Freeman, and Frédo Durand. Style transfer for headshot por-

traits. *ACM Transactions on Graphics (TOG)*, 33(4):148, 2014.

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

[32] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. In *European Conference on Computer Vision*, pages 349–365. Springer, 2018.