Demystifying Neural Style Transfer

Yanghao Li[†] **Naiyan Wang**[‡] **Jiaying Liu**[†*] **Xiaodi Hou**[‡]

[†] Institute of Computer Science and Technology, Peking University

[‡] TuSimple

lyttonhao@pku.edu.cn   winsty@gmail.com   liujiaying@pku.edu.cn   xiaodi.hou@gmail.com

Abstract

Neural Style Transfer [Gatys *et al.*, 2016] has recently demonstrated very exciting results which catches eyes in both academia and industry. Despite the amazing results, the principle of neural style transfer, especially why the Gram matrices could represent style remains unclear. In this paper, we propose a novel interpretation of neural style transfer by treating it as a domain adaptation problem. Specifically, we theoretically show that matching the Gram matrices of feature maps is equivalent to minimize the Maximum Mean Discrepancy (MMD) with the second order polynomial kernel. Thus, we argue that the essence of neural style transfer is to match the feature distributions between the style images and the generated images. To further support our standpoint, we experiment with several other distribution alignment methods, and achieve appealing results. We believe this novel interpretation connects these two important research fields, and could enlighten future researches.

1 Introduction

Transferring the style from one image to another image is an interesting yet difficult problem. There have been many efforts to develop efficient methods for automatic style transfer [Hertzmann *et al.*, 2001; Efros and Freeman, 2001; Efros and Leung, 1999; Shih *et al.*, 2014; Kwatra *et al.*, 2005]. Recently, Gatys *et al.* proposed a seminal work [Gatys *et al.*, 2016]: It captures the style of artistic images and transfer it to other images using Convolutional Neural Networks (CNN). This work formulated the problem as finding an image that matching both the content and style statistics based on the neural activations of each layer in CNN. It achieved impressive results and several follow-up works improved upon this innovative approaches [Johnson *et al.*, 2016; Ulyanov *et al.*, 2016; Ruder *et al.*, 2016; Ledig *et al.*, 2016]. Despite the fact that this work has drawn lots of attention, the fundamental element of style representation: the Gram matrix in [Gatys *et al.*, 2016] is not fully explained. The reason

why Gram matrix can represent artistic style still remains a mystery.

In this paper, we propose a novel interpretation of neural style transfer by casting it as a special domain adaptation [Beijbom, 2012; Patel *et al.*, 2015] problem. We theoretically prove that matching the Gram matrices of the neural activations can be seen as minimizing a specific Maximum Mean Discrepancy (MMD) [Gretton *et al.*, 2012a]. This reveals that neural style transfer is intrinsically a process of distribution alignment of the neural activations between images. Based on this illuminating analysis, we also experiment with other distribution alignment methods, including MMD with different kernels and a simplified moment matching method. These methods achieve diverse but all reasonable style transfer results. Specifically, a transfer method by MMD with linear kernel achieves comparable visual results yet with a lower complexity. Thus, the second order interaction in Gram matrix is not a must for style transfer. Our interpretation provides a promising direction to design style transfer methods with different visual results. To summarize, our contributions are shown as follows:

1. First, we demonstrate that matching Gram matrices in neural style transfer [Gatys *et al.*, 2016] can be reformulated as minimizing MMD with the second order polynomial kernel.

2. Second, we extend the original neural style transfer with different distribution alignment methods based on our novel interpretation.

2 Related Work

In this section, we briefly review some closely related works and the key concept MMD in our interpretation.

Style Transfer Style transfer is an active topic in both academia and industry. Traditional methods mainly focus on the non-parametric patch-based texture synthesis and transfer, which resamples pixels or patches from the original source texture images [Hertzmann *et al.*, 2001; Efros and Freeman, 2001; Efros and Leung, 1999; Liang *et al.*, 2001]. Different methods were proposed to improve the quality of the patch-based synthesis and constrain the structure of the target image. For example, the image quilting algorithm based on dynamic programming was proposed to find optimal texture

---

*Corresponding author

boundaries in [Efros and Freeman, 2001]. A Markov Random Field (MRF) was exploited to preserve global texture structures in [Frigo *et al.*, 2016]. However, these non-parametric methods suffer from a fundamental limitation that they only use the low-level features of the images for transfer.

Recently, neural style transfer [Gatys *et al.*, 2016] has demonstrated remarkable results for image stylization. It fully takes the advantage of the powerful representation of Deep Convolutional Neural Networks (CNN). This method used Gram matrices of the neural activations from different layers of a CNN to represent the artistic style of a image. Then it used an iterative optimization method to generate a new image from white noise by matching the neural activations with the content image and the Gram matrices with the style image. This novel technique attracts many follow-up works for different aspects of improvements and applications. To speed up the iterative optimization process in [Gatys *et al.*, 2016], Johnson *et al.* [Johnson *et al.*, 2016] and Ulyanov *et al.* [Ulyanov *et al.*, 2016] trained a feed-forward generative network for fast neural style transfer. To improve the transfer results in [Gatys *et al.*, 2016], different complementary schemes are proposed, including spatial constraints [Selim *et al.*, 2016], semantic guidance [Champandard, 2016] and Markov Random Field (MRF) prior [Li and Wand, 2016]. There are also some extension works to apply neural style transfer to other applications. Ruder *et al.* [Ruder *et al.*, 2016] incorporated temporal consistence terms by penalizing deviations between frames for video style transfer. Selim *et al.* [Selim *et al.*, 2016] proposed novel spatial constraints through gain map for portrait painting transfer. Although these methods further improve over the original neural style transfer, they all ignore the fundamental question in neural style transfer: *Why could the Gram matrices represent the artistic style?* This vagueness of the understanding limits the further research on the neural style transfer.

Domain Adaptation   Domain adaptation belongs to the area of transfer learning [Pan and Yang, 2010]. It aims to transfer the model that is learned on the source domain to the unlabeled target domain. The key component of domain adaptation is to measure and minimize the difference between source and target distributions. The most common discrepancy metric is Maximum Mean Discrepancy (MMD) [Gretton *et al.*, 2012a], which measure the difference of sample mean in a Reproducing Kernel Hilbert Space. It is a popular choice in domain adaptation works [Tzeng *et al.*, 2014; Long *et al.*, 2015; Long *et al.*, 2016]. Besides MMD, Sun *et al.* [Sun *et al.*, 2016] aligned the second order statistics by whitening the data in source domain and then re-correlating to the target domain. In [Li *et al.*, 2017], Li *et al.* proposed a parameter-free deep adaptation method by simply modulating the statistics in all Batch Normalization (BN) layers.

Maximum Mean Discrepancy   Suppose there are two sets of samples $X = \{\mathbf{x}_i\}_{i=1}^n$ and $Y = \{\mathbf{y}_j\}_{j=1}^m$ where $\mathbf{x}_i$ and $\mathbf{y}_j$ are generated from distributions p and q, respectively. Maximum Mean Discrepancy (MMD) is a popular test statistic for the two-sample testing problem, where acceptance or rejection decisions are made for a null hypothesis $p = q$ [Gretton

et al., 2012a]. Since the population MMD vanishes if and only $p = q$, the MMD statistic can be used to measure the difference between two distributions. Specifically, we calculates MMD defined by the difference between the mean embedding on the two sets of samples. Formally, the squared MMD is defined as:

$$
\begin{aligned}
&\mathrm{MMD}^2[X, Y] \\
&= \|\mathbf{E}_x[\phi(\mathbf{x})] - \mathbf{E}_y[\phi(\mathbf{y})]\|^2 \\
&= \|\frac{1}{n}\sum_{i=1}^n \phi(\mathbf{x}_i) - \frac{1}{m}\sum_{j=1}^m \phi(\mathbf{y}_j)\|^2 \\
&= \frac{1}{n^2}\sum_{i=1}^n\sum_{i'=1}^n \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_{i'}) + \frac{1}{m^2}\sum_{j=1}^m\sum_{j'=1}^m \phi(\mathbf{y}_j)^T\phi(\mathbf{y}_{j'}) \\
&\quad - \frac{2}{nm}\sum_{i=1}^n\sum_{j=1}^m \phi(\mathbf{x}_i)^T\phi(\mathbf{y}_j),
\end{aligned}
\tag{1}
$$

where $\phi(\cdot)$ is the explicit feature mapping function of MMD. Applying the associated kernel function $k(\mathbf{x}, \mathbf{y}) = \langle\phi(\mathbf{x}), \phi(\mathbf{y})\rangle$, the Eq. 1 can be expressed in the form of kernel:

$$
\begin{aligned}
&\mathrm{MMD}^2[X, Y] \\
&= \frac{1}{n^2}\sum_{i=1}^n\sum_{i'=1}^n k(\mathbf{x}_i, \mathbf{x}_{i'}) + \frac{1}{m^2}\sum_{j=1}^m\sum_{j'=1}^m k(\mathbf{y}_j, \mathbf{y}_{j'}) \\
&\quad - \frac{2}{nm}\sum_{i=1}^n\sum_{j=1}^m k(\mathbf{x}_i, \mathbf{y}_j).
\end{aligned}
\tag{2}
$$

The kernel function $k(\cdot, \cdot)$ implicitly defines a mapping to a higher dimensional feature space.

3   Understanding Neural Style Transfer

In this section, we first theoretically demonstrate that matching Gram matrices is equivalent to minimizing a specific form of MMD. Then based on this interpretation, we extend the original neural style transfer with different distribution alignment methods.

Before explaining our observation, we first briefly review the original neural style transfer approach [Gatys *et al.*, 2016]. The goal of style transfer is to generate a stylized image $\mathbf{x}^*$ given a content image $\mathbf{x}_c$ and a reference style image $\mathbf{x}_s$. The feature maps of $\mathbf{x}^*$, $\mathbf{x}_c$ and $\mathbf{x}_s$ in the layer l of a CNN are denoted by $\mathbf{F}^l \in \mathbb{R}^{N_l \times M_l}$, $\mathbf{P}^l \in \mathbb{R}^{N_l \times M_l}$ and $\mathbf{S}^l \in \mathbb{R}^{N_l \times M_l}$ respectively, where N_l is the number of the feature maps in the layer l and M_l is the height times the width of the feature map.

In [Gatys *et al.*, 2016], neural style transfer iteratively generates $\mathbf{x}^*$ by optimizing a content loss and a style loss:

$$
\mathcal{L} = \alpha\mathcal{L}_{content} + \beta\mathcal{L}_{style},
\tag{3}
$$

where α and β are the weights for content and style losses, $\mathcal{L}_{content}$ is defined by the squared error between the feature maps of a specific layer l for $\mathbf{x}^*$ and $\mathbf{x}_c$:

$$
\mathcal{L}_{content} = \frac{1}{2}\sum_{i=1}^{N_l}\sum_{j=1}^{M_l}(F_{ij}^l - P_{ij}^l)^2,
\tag{4}
$$

and $\mathcal{L}_{style}$ is the sum of several style loss $\mathcal{L}_{style}^l$ in different layers:

$$\mathcal{L}_{style} = \sum_l w_l \mathcal{L}_{style}^l, \qquad (5)$$

where w_l is the weight of the loss in the layer l and $\mathcal{L}_{style}^l$ is defined by the squared error between the features correlations expressed by Gram matrices of $\mathbf{x}^*$ and $\mathbf{x}_s$:

$$\mathcal{L}_{style}^l = \frac{1}{4N_l^2 M_l^2} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} (G_{ij}^l - A_{ij}^l)^2, \qquad (6)$$

where the Gram matrix $\mathbf{G}^l \in \mathbb{R}^{N_l \times N_l}$ is the inner product between the vectorized feature maps of $\mathbf{x}^*$ in layer l:

$$G_{ij}^l = \sum_{k=1}^{M_l} F_{ik}^l F_{jk}^l, \qquad (7)$$

and similarly $\mathbf{A}^l$ is the Gram matrix corresponding to $\mathbf{S}^l$.

3.1 Reformulation of the Style Loss

In this section, we reformulated the style loss $\mathcal{L}_{style}$ in Eq. 6. By expanding the Gram matrix in Eq. 6, we can get the formulation of Eq. 8, where $\mathbf{f}_{\cdot k}^l$ and $\mathbf{s}_{\cdot k}^l$ is the k-th column of $\mathbf{F}^l$ and $\mathbf{S}^l$.

By using the second order degree polynomial kernel $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y})^2$, Eq. 8 can be represented as:

$$\begin{aligned}
\mathcal{L}_{style}^l =& \frac{1}{4N_l^2 M_l^2} \sum_{k_1=1}^{M_l} \sum_{k_2=1}^{M_l} \Big(k(\mathbf{f}_{\cdot k_1}^l, \mathbf{f}_{\cdot k_2}^l) \\
&+ k(\mathbf{s}_{\cdot k_1}^l, \mathbf{s}_{\cdot k_2}^l) - 2k(\mathbf{f}_{\cdot k_1}^l, \mathbf{s}_{\cdot k_2}^l) \Big) \\
=& \frac{1}{4N_l^2} \mathrm{MMD}^2[\mathcal{F}^l, \mathcal{S}^l],
\end{aligned} \qquad (9)$$

where $\mathcal{F}^l$ is the feature set of $\mathbf{x}^*$ where each sample is a column of $\mathbf{F}^l$, and $\mathcal{S}^l$ corresponds to the style image $\mathbf{x}_s$. In this way, the activations at each position of feature maps is considered as an individual sample. Consequently, the style loss ignores the positions of the features, which is desired for style transfer. In conclusion, the above reformulations suggest two important findings:

1. The style of a image can be intrinsically represented by feature distributions in different layers of a CNN.

2. The style transfer can be seen as a distribution alignment process from the content image to the style image.

3.2 Different Adaptation Methods for Neural Style Transfer

Our interpretation reveals that neural style transfer can be seen as a problem of distribution alignment, which is also at the core in domain adaptation. If we consider the style of one image in a certain layer of CNN as a "domain", style transfer can also be seen as a special domain adaptation problem. The specialty of this problem lies in that we treat the feature at each position of feature map as one individual data sample, instead of that in traditional domain adaptation problem

in which we treat each image as one data sample. (*e.g.* The feature map of the last convolutional layer in VGG-19 model is of size 14×14, then we have totally 196 samples in this "domain".)

Inspired by the studies of domain adaptation, we extend neural style transfer with different adaptation methods in this subsection.

MMD with Different Kernel Functions As shown in Eq. 9, matching Gram matrices in neural style transfer can been seen as a MMD process with second order polynomial kernel. It is very natural to apply other kernel functions for MMD in style transfer. First, if using MMD statistics to measure the style discrepancy, the style loss can be defined as:

$$\begin{aligned}
\mathcal{L}_{style}^l =& \frac{1}{Z_k^l} \mathrm{MMD}^2[\mathcal{F}^l, \mathcal{S}^l], \\
=& \frac{1}{Z_k^l} \sum_{i=1}^{M_l} \sum_{j=1}^{M_l} \Big(k(\mathbf{f}_{\cdot i}^l, \mathbf{f}_{\cdot j}^l) + k(\mathbf{s}_{\cdot i}^l, \mathbf{s}_{\cdot j}^l) - 2k(\mathbf{f}_{\cdot i}^l, \mathbf{s}_{\cdot j}^l) \Big),
\end{aligned}$$

$$(10)$$

where Z_k^l is the normalization term corresponding to different scale of the feature map in the layer l and the choice of kernel function. Theoretically, different kernel function implicitly maps features to different higher dimensional space. Thus, we believe that different kernel functions should capture different aspects of a style. We adopt the following three popular kernel functions in our experiments:

(1) Linear kernel: $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$;

(2) Polynomial kernel: $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + c)^d$;

(3) Gaussian kernel: $k(\mathbf{x}, \mathbf{y}) = \exp\big(-\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{2\sigma^2} \big)$.

For polynomial kernel, we only use the version with $d = 2$. Note that matching Gram matrices is equivalent to the polynomial kernel with $c = 0$ and $d = 2$. For the Gaussian kernel, we adopt the unbiased estimation of MMD [Gretton *et al.*, 2012b], which samples M_l pairs in Eq. 10 and thus can be computed with linear complexity.

BN Statistics Matching In [Li *et al.*, 2017], the authors found that the statistics (*i.e.* mean and variance) of Batch Normalization (BN) layers contains the traits of different domains. Inspired by this observation, they utilized separate BN statistics for different domain. This simple operation aligns the different domain distributions effectively. As a special domain adaptation problem, we believe that BN statistics of a certain layer can also represent the style. Thus, we construct another style loss by aligning the BN statistics (mean and standard deviation) of two feature maps between two images:

$$\mathcal{L}_{style}^l = \frac{1}{N_l} \sum_{i=1}^{N_l} \Big((\mu_{F^l}^i - \mu_{S^l}^i)^2 + (\sigma_{F^l}^i - \sigma_{S^l}^i)^2 \Big), \quad (11)$$

where $\mu_{F^l}^i$ and $\sigma_{F^l}^i$ is the mean and standard deviation of the i-th feature channel among all the positions of the feature map

$$\mathcal{L}_{style}^l = \frac{1}{4N_l^2 M_l^2} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} (\sum_{k=1}^{M_l} F_{ik}^l F_{jk}^l - \sum_{k=1}^{M_l} S_{ik}^l S_{jk}^l)^2$$

$$= \frac{1}{4N_l^2 M_l^2} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} \left((\sum_{k=1}^{M_l} F_{ik}^l F_{jk}^l)^2 + (\sum_{k=1}^{M_l} S_{ik}^l S_{jk}^l)^2 - 2(\sum_{k=1}^{M_l} F_{ik}^l F_{jk}^l)(\sum_{k=1}^{M_l} S_{ik}^l S_{jk}^l) \right)$$

$$= \frac{1}{4N_l^2 M_l^2} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} \sum_{k_1=1}^{M_l} \sum_{k_2=1}^{M_l} (F_{ik_1}^l F_{jk_1}^l F_{ik_2}^l F_{jk_2}^l + S_{ik_1}^l S_{jk_1}^l S_{ik_2}^l S_{jk_2}^l - 2F_{ik_1}^l F_{jk_1}^l S_{ik_2}^l S_{jk_2}^l)$$

$$= \frac{1}{4N_l^2 M_l^2} \sum_{k_1=1}^{M_l} \sum_{k_2=1}^{M_l} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} (F_{ik_1}^l F_{jk_1}^l F_{ik_2}^l F_{jk_2}^l + S_{ik_1}^l S_{jk_1}^l S_{ik_2}^l S_{jk_2}^l - 2F_{ik_1}^l F_{jk_1}^l S_{ik_2}^l S_{jk_2}^l) \tag{8}$$

$$= \frac{1}{4N_l^2 M_l^2} \sum_{k_1=1}^{M_l} \sum_{k_2=1}^{M_l} \left((\sum_{i=1}^{N_l} F_{ik_1}^l F_{ik_2}^l)^2 + (\sum_{i=1}^{N_l} S_{ik_1}^l S_{ik_2}^l)^2 - 2(\sum_{i=1}^{N_l} F_{ik_1}^l S_{ik_2}^l)^2 \right)$$

$$= \frac{1}{4N_l^2 M_l^2} \sum_{k_1=1}^{M_l} \sum_{k_2=1}^{M_l} \left((\mathbf{f}_{\cdot k_1}^l{}^T \mathbf{f}_{\cdot k_2}^l)^2 + (\mathbf{s}_{\cdot k_1}^l{}^T \mathbf{s}_{\cdot k_2}^l)^2 - 2(\mathbf{f}_{\cdot k_1}^l{}^T \mathbf{s}_{\cdot k_2}^l)^2 \right),$$

in the layer l for image $\mathbf{x}^*$:

$$\mu_{F^l}^i = \frac{1}{M_l} \sum_{j=1}^{M_l} F_{ij}^l, \quad \sigma_{F^l}^i{}^2 = \frac{1}{M_l} \sum_{j=1}^{M_l} (F_{ij}^l - \mu_{F^l}^i)^2, \quad (12)$$

and $\mu_{S^l}^i$ and $\sigma_{S^l}^i$ correspond to the style image $\mathbf{x}_s$.

The aforementioned style loss functions are all differentiable and thus the style matching problem can be solved by back propagation iteratively.

4 Results

In this section, we briefly introduce some implementation details and present results by our extended neural style transfer methods. Furthermore, we also show the results of fusing different neural style transfer methods, which combine different style losses. In the following, we refer the four extended style transfer methods introduced in Sec. 3.2 as *linear*, *poly*, *Gaussian* and *BN*, respectively. The images in the experiments are collected from the public implementations of neural style transfer[1][2][3].

Implementation Details In the implementation, we use the VGG-19 network [Simonyan and Zisserman, 2015] following the choice in [Gatys *et al.*, 2016]. We also adopt the *relu4_2* layer for the content loss, and *relu1_1*, *relu2_1*, *relu3_1*, *relu4_1*, *relu5_1* for the style loss. The default weight factor w_l is set as 1.0 if it is not specified. The target image $\mathbf{x}^*$ is initialized randomly and optimized iteratively until the relative change between successive iterations is under 0.5%. The maximum number of iterations is set as 1000. For the method with Gaussian kernel MMD, the kernel bandwidth σ^2 is fixed as the mean of squared l_2 distances of the sampled pairs since

it does not affect a lot on the visual results. Our implementation is based on the MXNet [Chen *et al.*, 2016] implementation[1] which reproduces the results of original neural style transfer [Gatys *et al.*, 2016].

Since the scales of the gradients of the style loss differ for different methods, and the weights α and β in Eq. 3 affect the results of style transfer, we fix some factors to make a fair comparison. Specifically, we set $\alpha = 1$ because the content losses are the same among different methods. Then, for each method, we first manually select a proper β' such that the gradients on the $\mathbf{x}^*$ from the style loss are of the same order of magnitudes as those from the content loss. Thus, we can manipulate a balance factor γ ($\beta = \gamma\beta'$) to make trade-off between the content and style matching.

4.1 Different Style Representations
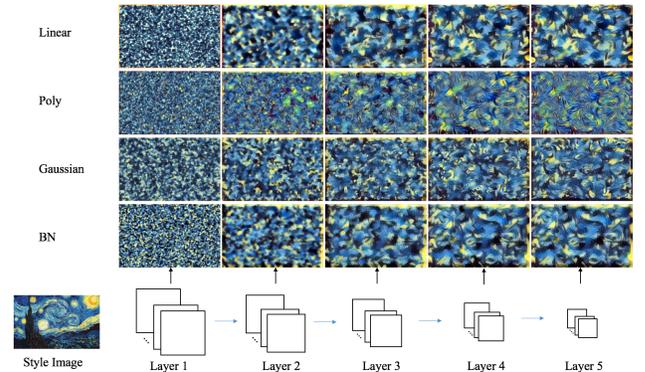
Figure 1: Style reconstructions of different methods in five layers, respectively. Each row corresponds to one method and the reconstruction results are obtained by only using the style loss $\mathcal{L}_{style}$ with $\alpha = 0$. We also reconstruct different style representations in different subsets of layers of VGG network. For example, layer 3 contains the style loss of the first 3 layers ($w_1 = w_2 = w_3 = 1.0$ and $w_4 = w_5 = 0.0$).

To validate that the extended neural style transfer methods can capture the style representation of an artistic image,

---

[1]https://github.com/dmlc/mxnet/tree/master/example/neural-style

[2]https://github.com/jcjohnson/neural-style

[3]https://github.com/jcjohnson/fast-neural-style

(a) Content / Style   (b) $\gamma = 0.1$ (c) $\gamma = 0.2$ (d) $\gamma = 1.0$ (e) $\gamma = 5.0$ (f) $\gamma = 10.0$
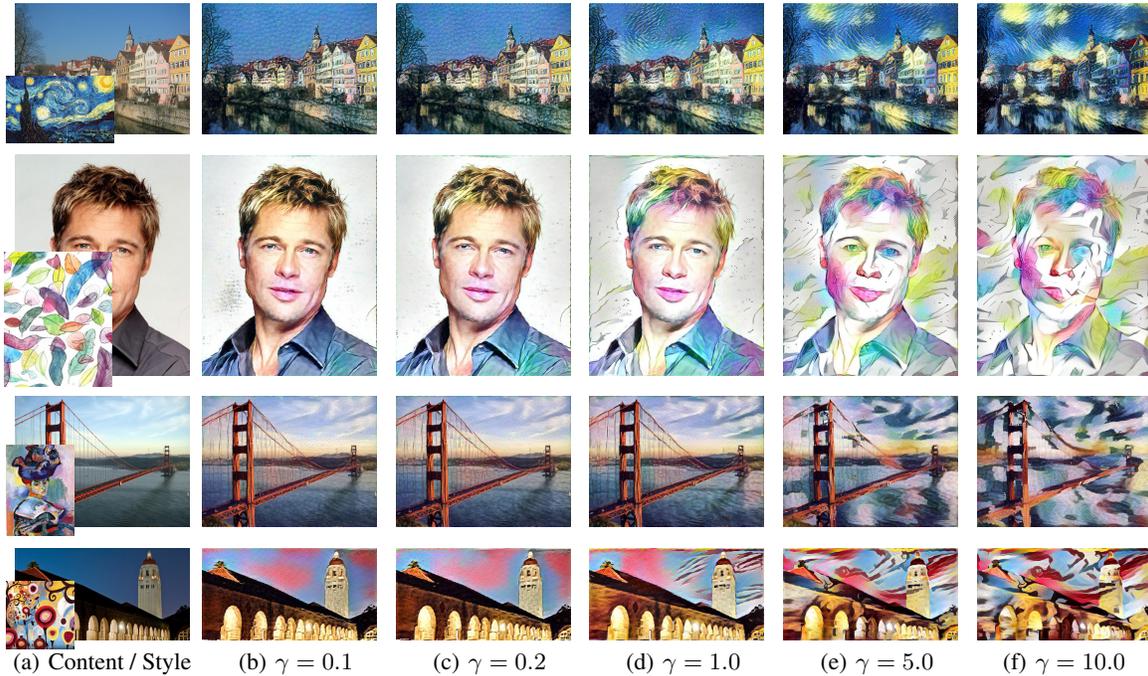
Figure 2: Results of the four methods (*linear*, *poly*, *Gaussian* and *BN*) with different balance factor γ. Larger γ means more emphasis on the style loss.

we first visualize the style reconstruction results of different methods only using the style loss in Fig. 1. Moreover, Fig. 1 also compares the style representations of different layers. On one hand, for a specific method (one row), the results show that different layers capture different levels of style: The textures in the top layers usually has larger granularity than those in the bottom layers. This is reasonable because each neuron in the top layers has larger receptive field and thus has the ability to capture more global textures. On the other hand, for a specific layer, Fig. 1 also demonstrates that the style captured by different methods differs. For example, in top layers, the textures captured by MMD with a linear kernel are composed by thick strokes. Contrarily, the textures captured by MMD with a polynomial kernel are more fine grained.

4.2 Result Comparisons

Effect of the Balance Factor　We first explore the effect of the balance factor between the content loss and style loss by varying the weight γ. Fig. 2 shows the results of four transfer methods with various γ from 0.1 to 10.0. As intended, the global color information in the style image is successfully transferred to the content image, and the results with smaller γ preserve more content details as shown in Fig. 2(b) and Fig. 2(c). When γ becomes larger, more stylized textures are incorporated into the results. For example, Fig. 2(e) and Fig. 2(f) have much more similar illumination and textures with the style image, while Fig. 2(d) shows a balanced result between the content and style. Thus, users can make trade-off between the content and the style by varying γ.

(a) Content / Style   (b) linear   (c) poly   (d) Gaussian   (e) BN
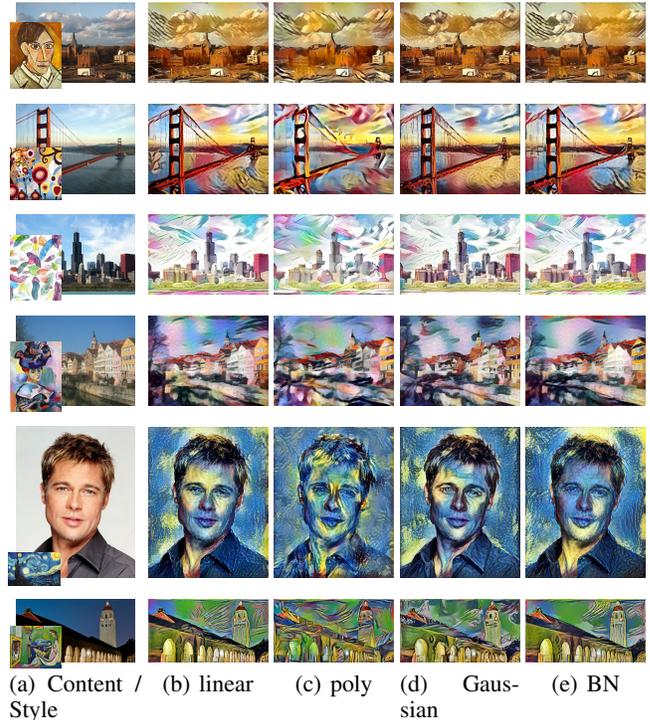
Figure 3: Visual results of several style transfer methods, including *linear*, *poly*, *Gaussian* and *BN*. The balance factors γ in the six examples are 2.0, 2.0, 2.0, 5.0, 5.0 and 5.0, respectively.

Figure 4: Results of two fusion methods: *BN + poly* and *linear + Gaussian*. The top two rows are the results of first fusion method and the bottom two rows correspond to the second one. Each column shows the results of a balance weight between the two methods. γ is set as 5.0.

Comparisons of Different Transfer Methods Fig. 3 presents the results of various pairs of content and style images with different transfer methods[4]. Similar to matching Gram matrices, which is equivalent to the *poly* method, the other three methods can also transfer satisfied styles from the specified style images. This empirically demonstrates the correctness of our interpretation of neural style transfer: Style transfer is essentially a domain adaptation problem, which aligns the feature distributions. Particularly, when the weight on the style loss becomes higher (namely, larger γ), the differences among the four methods are getting larger. This indicates that these methods implicitly capture different aspects of style, which has also been shown in Fig. 1. Since these methods have their unique properties, they could provide more choices for users to stylize the content image. For example, *linear* achieves comparable results with other methods, yet requires lower computation complexity.

Fusion of Different Neural Style Transfer Methods Since we have several different neural style transfer methods, we propose to combine them to produce new transfer results. Fig. 4 demonstrates the fusion results of two combinations (*linear + Gaussian* and *poly + BN*). Each row presents the results with different balance between the two methods. For example, Fig. 4(b) in the first two rows emphasize more on *BN* and Fig. 4(f) emphasizes more on *poly*. The results in

the middle columns show the interpolation between these two methods. We can see that the styles of different methods are blended well using our method.

5   Conclusion

Despite the great success of neural style transfer, the rationale behind neural style transfer was far from crystal. The vital "trick" for style transfer is to match the Gram matrices of the features in a layer of a CNN. Nevertheless, subsequent literatures about neural style transfer just directly improves upon it without investigating it in depth. In this paper, we present a timely explanation and interpretation for it. First, we theoretically prove that matching the Gram matrices is equivalent to a specific Maximum Mean Discrepancy (MMD) process. Thus, the style information in neural style transfer is intrinsically represented by the distributions of activations in a CNN, and the style transfer can be achieved by distribution alignment. Moreover, we exploit several other distribution alignment methods, and find that these methods all yield promising transfer results. Thus, we justify the claim that neural style transfer is essentially a special domain adaptation problem both theoretically and empirically. We believe this interpretation provide a new lens to re-examine the style transfer problem, and will inspire more exciting works in this research area.

Acknowledgement

---

[4]More results can be found at
http://www.icst.pku.edu.cn/struct/Projects/mmdstyle/result-1000/show-full.html

References

[Beijbom, 2012] Oscar Beijbom. Domain adaptations for computer vision applications. *arXiv preprint arXiv:1211.4860*, 2012.

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

[Chen *et al.*, 2016] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. *NIPS Workshop on Machine Learning Systems*, 2016.

[Efros and Freeman, 2001] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, 2001.

[Efros and Leung, 1999] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *ICCV*, 1999.

[Frigo *et al.*, 2016] Oriel Frigo, Neus Sabater, Julie Delon, and Pierre Hellier. Split and match: Example-based adaptive patch sampling for unsupervised style transfer. In *CVPR*, 2016.

[Gatys *et al.*, 2016] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

[Gretton *et al.*, 2012a] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[Gretton *et al.*, 2012b] Arthur Gretton, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, Kenji Fukumizu, and Bharath K Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *NIPS*, 2012.

[Hertzmann *et al.*, 2001] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *SIGGRAPH*, 2001.

[Johnson *et al.*, 2016] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[Kwatra *et al.*, 2005] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Transactions on Graphics*, 24(3):795–802, 2005.

[Ledig *et al.*, 2016] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[Li and Wand, 2016] Chuan Li and Michael Wand. Combining Markov random fields and convolutional neural networks for image synthesis. In *CVPR*, 2016.

[Li *et al.*, 2017] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *ICLRW*, 2017.

[Liang *et al.*, 2001] 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*, 20(3):127–150, 2001.

[Long *et al.*, 2015] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[Long *et al.*, 2016] Mingsheng Long, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, 2016.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[Patel *et al.*, 2015] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, 32(3):53–69, 2015.

[Ruder *et al.*, 2016] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *GCPR*, 2016.

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

[Shih *et al.*, 2014] YiChang Shih, Sylvain Paris, Connelly Barnes, William T Freeman, and Frédo Durand. Style transfer for headshot portraits. *ACM Transactions on Graphics*, 33(4):148, 2014.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[Sun *et al.*, 2016] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. *AAAI*, 2016.

[Tzeng *et al.*, 2014] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[Ulyanov *et al.*, 2016] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, 2016.