# Style Transfer by Relaxed Optimal Transport and Self-Similarity

Nicholas Kolkin[1] Jason Salavon[2] Gregory Shakhnarovich[1]
[1]Toyota Technological Institute at Chicago [2]University of Chicago
nick.kolkin@ttic.edu, salavon@uchicago.edu, greg@ttic.edu

## Abstract

*The goal of style transfer algorithms is to render the content of one image using the style of another. We propose Style Transfer by Relaxed Optimal Transport and Self-Similarity (STROTSS), a new optimization-based style transfer algorithm. We extend our method to allow user-specified point-to-point or region-to-region control over visual similarity between the style image and the output. Such guidance can be used to either achieve a particular visual effect or correct errors made by unconstrained style transfer. In order to quantitatively compare our method to prior work, we conduct a large-scale user study designed to assess the style-content tradeoff across settings in style transfer algorithms. Our results indicate that for any desired level of content preservation, our method provides higher quality stylization than prior work.*

## 1 Introduction

One of the main challenges of style transfer is formalizing 'content' and 'style', terms which evoke strong intuitions but are hard to even define semantically. We propose formulations of each term which are novel in the domain of style transfer, but have a long history of successful application in computer vision more broadly. We hope that related efforts to refine definitions of both style and content will eventually lead to more robust recognition systems, but in this work we solely focus on their utility for style transfer.

We define style as a distribution over features extracted by a deep neural network, and measure the distance between these distributions using an efficient approximation of the Earth Movers Distance initially proposed in the Natural Language Processing community [14]. This definition of style similarity is not only well motivated statistically, but also intuitive. The goal of style transfer is to deploy the visual attributes of the style image onto the content image with minimum distortion to the content's underlying layout and semantics; in essence to 'optimally transport' these visual attributes.

Our definition of content is inspired by the concept of self-similarity, and the notion that human perceptual system is robust because it identifies objects based on their appearance relative to their surroundings, rather than absolute appearance. Defining content similarity in this way disconnects the term somewhat from pixels precise values making it easier to satisfy than the definitions used in prior work. This allows the output of our algorithm to maintain the perceived semantics and spatial layout of the content image,

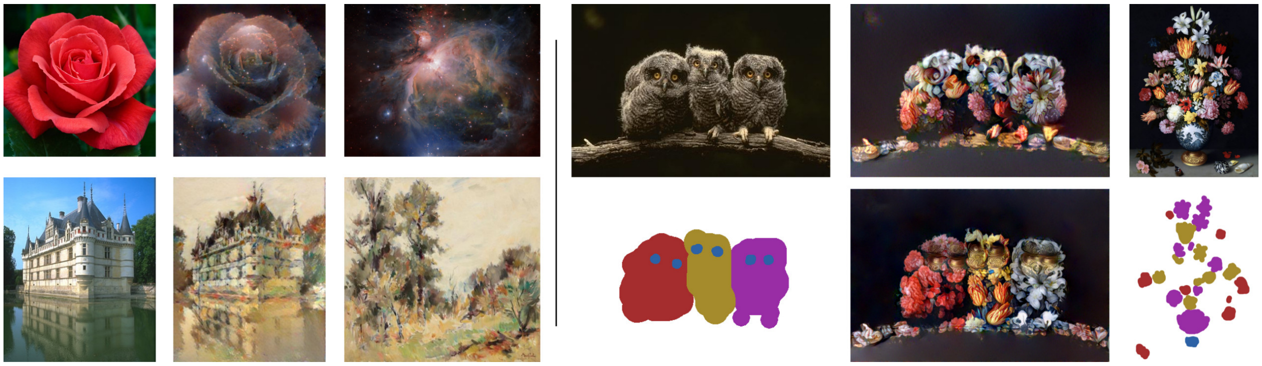


Figure 1: Examples of our output for unconstrained (left) and guided (right) style transfer.Images are arranged in order of content, output, style. Below the content and style image on the right we visualize the user-defined region-to-region guidance used to generate the output in the middle.

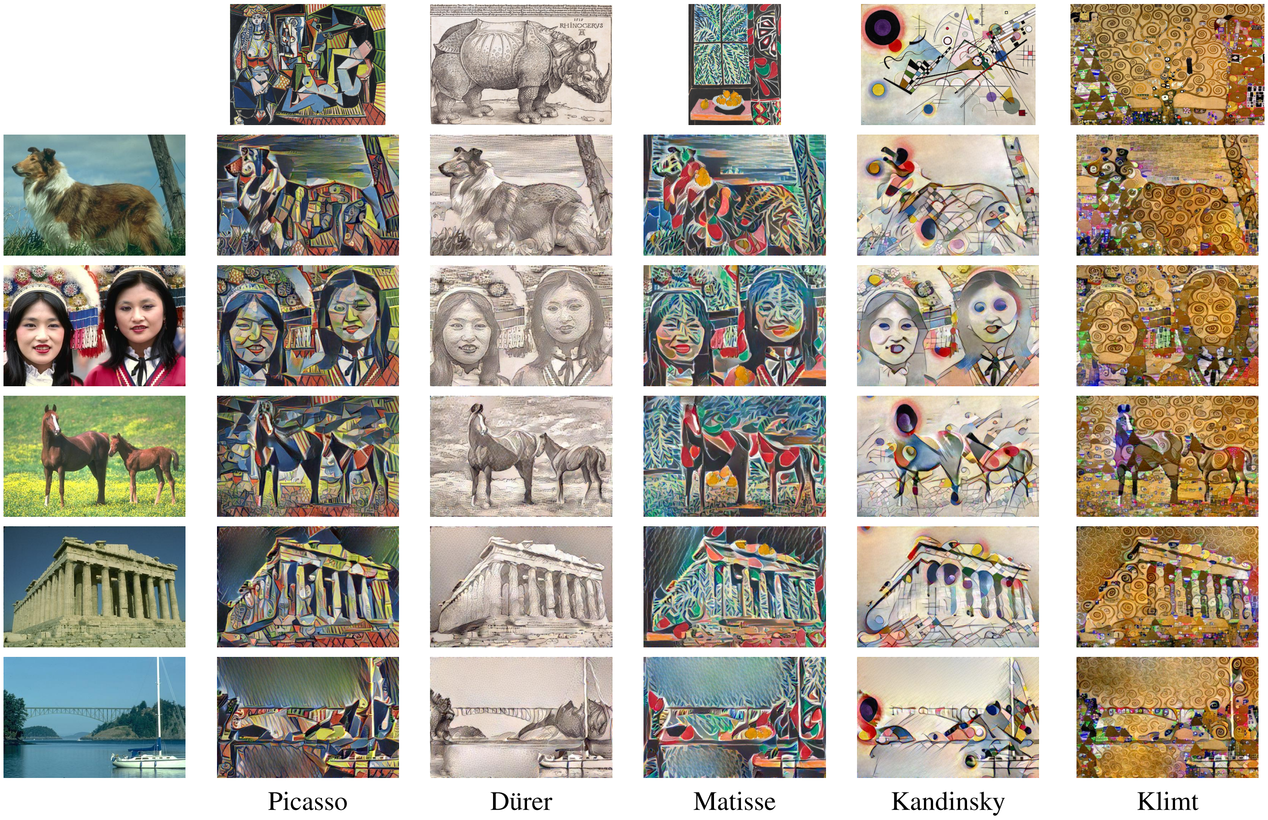


Figure 2: Examples of the effect of different content images on the same style, and vice-versa

while drastically differing in pixel space.

To increase utility of style transfer as an artistic tool, it is important that users can easily and intuitively control the algorithm's output. We extend our formulation to allow region-to-region constraints on style transfer (e.g., ensuring that hair in the content image is stylized using clouds in the style image) and point-to-point constraints (e.g., ensuring that the eye in the content image is stylized in the same way as the eye in a painting).

We quantitatively compare our method to prior work using human evaluations gathered from 662 workers on Amazon Mechanical Turk (AMT). Workers evaluated content preservation and stylization quality separately. Workers were shown two algorithms' output for the same inputs in addition to either the content or style input, then asked which has more similar content or style respectively to the displayed input. In this way are able to quantify the performance of each algorithm along both axes. By evaluate our method and prior work for multiple hyper-parameter settings, we also measure the trade-off within each method between stylization and content preservation as hyper-parameters change. Our results indicate that for any desired level of content preservation, our method provides higher quality stylization than prior work.

## 2 Methods

Like the original Neural Style Transfer algorithm proposed by Gatys et al. [4] our method takes two inputs, a style image $I_S$ and a content image $I_C$, and uses the gradient descent variant RMSprop [11] to minimize our proposed objective function (equation 1) with respect to the output image $X$.

$$L(X, I_C, I_S) = \frac{\alpha \ell_C + \ell_m + \ell_r + \frac{1}{\alpha}\ell_p}{2 + \alpha + \frac{1}{\alpha}} \tag{1}$$

We describe the content term of our loss $\alpha \ell_C$ in Section 2.2, and the style term $\ell_m + \ell_r + \frac{1}{\alpha}\ell_p$ in Section 2.3. The hyper-parameter $\alpha$ represents the relative importance of content preservation to stylization. Our method is iterative; let $X^{(t)}$ refer to the stylized output image at timestep $t$. We describe our initialization $X^{(0)}$ in Section 2.5.

### 2.1 Feature Extraction

Both our style and content loss terms rely upon extracting a rich feature representation from an arbitrary spatial location. In this work we use hypercolumns [21, 8] extracted from a subset of layers of VGG16 trained on ImageNet [26]. Let $\Phi(X)_i$ be the tensor of feature activations extracted from input image $X$ by layer $i$ of network $\Phi$. Given the set of layer indices $l_1, .., l_L$ we use bilinear upsampling to match the spatial dimensions of $\Phi(X)_{l_1} ... \Phi(X)_{l_L}$ to those of the original image $(X)$, then concatenate all such tensors along the feature dimension. This yields a hypercolumn at each pixel, that includes features which capture low-level edge and color features, mid-level texture features, and high-level semantic features [27]. For all experiments we use all convolutional layers of VGG16 except layers 9,10,12, and 13, which we exclude because of memory constraints.

## 2.2 Style Loss

Let $A = \{A_1, \ldots, A_n\}$ be a set of $n$ feature vectors extracted from $X^{(t)}$, and $B = \{B_1, \ldots, B_m\}$ be a set of $m$ features extracted from style image $I_S$. The style loss is derived from the Earth Movers Distance (EMD)[1]:

$$\text{EMD}(A, B) = \min_{T \geq 0} \sum_{ij} T_{ij} C_{ij} \tag{2}$$

$$s.t. \sum_j T_{ij} = 1/m \tag{3}$$

$$\sum_i T_{ij} = 1/n \tag{4}$$

where $T$ is the 'transport matrix', which defines partial pairwise assignments, and $C$ is the 'cost matrix' which defines how far an element in $A$ is from an element in $B$. $\text{EMD}(A, B)$ captures the distance between sets $A$ and $B$, but finding the optimal $T$ costs $O(\max(m, n)^3)$, and is therefore untenable for gradient descent based style transfer (where it would need to be computed at each update step). Instead we will use the Relaxed EMD [14]. To define this we will use two auxiliary distances, essentially each is the EMD with only one of the constraints (3) or (4):

$$R_A(A, B) = \min_{T \geq 0} \sum_{ij} T_{ij} C_{ij} \quad s.t. \quad \sum_j T_{ij} = 1/m \tag{5}$$

$$R_B(A, B) = \min_{T \geq 0} \sum_{ij} T_{ij} C_{ij} \quad s.t. \quad \sum_i T_{ij} = 1/n \tag{6}$$

we can then define the relaxed earth movers distance as:

$$\ell_r = REMD(A, B) = \max(R_A(A, B), R_B(A, B)) \tag{7}$$

This is equivalent to:

$$\ell_r = \max\left(\frac{1}{n} \sum_i \min_j C_{ij}, \frac{1}{m} \sum_j \min_i C_{ij}\right) \tag{8}$$

Computing this is dominated by computing the cost matrix $C$. We compute the cost of transport (ground metric) from $A_i$ to $B_j$ as the cosince distance between the two feature vectors,

$$C_{ij} = D_{\cos}(A_i, B_j) = 1 - \frac{A_i \cdot B_j}{\|A_i\| \|B_j\|} \tag{9}$$

We experimented with using the Euclidean distance between vectors instead, but the results were significantly worse, see the supplement for examples.

While $\ell_r$ does a good job of transferring the structural forms of the source image to the target, the cosine distance ignores the magnitude of the feature vectors. In practice this leads to visual artifacts in the output, most notably over-/under-saturation. To combat this we add a moment matching loss:

$$\ell_m = \frac{1}{d} \|\mu_A - \mu_B\|_1 + \frac{1}{d^2} \|\Sigma_A - \Sigma_B\|_1 \tag{10}$$

where $\mu_A$, $\Sigma_A$ are the mean and covariance of the feature vectors in set $A$, and $\mu_B$ and $\Sigma_B$ are defined in the same way.

We also add a color matching loss, $\ell_p$ to encourage our output and the style image to have a similar palette. $\ell_p$ is defined using the Relaxed EMD between pixel colors in $X^{(t)}$ and $I_S$, this time and using Euclidean distance as a ground metric. We find it beneficial to convert the colors from RGB into a decorrelated colorspace with mean color as one channel when computing this term. Because palette shifting is at odds with content preservation, we weight this term by $\frac{1}{\alpha}$.

## 2.3 Content Loss

Our content loss is motivated by the observation that robust pattern recognition can be built using local self-similarity descriptors [25]. An every day example of this is the phenomenon called pareidolia, where the self-similarity patterns of inanimate objects are perceived as faces because they match a loose template. Formally, let $D^X$ be the pairwise cosine distance matrix of all (hypercolumn) feature vectors extracted from $X^{(t)}$, and let $D^{I_C}$ be defined analogously for the content image. We visualize several potential rows of $D^X$ in Figure 3. We define our content loss as:

$$\mathcal{L}_{content}(X, C) = \frac{1}{n^2} \sum_{i,j} \left| \frac{D^X_{ij}}{\sum_i D^X_{ij}} - \frac{D^{I_C}_{ij}}{\sum_i D^{I_C}_{ij}} \right| \tag{11}$$

In other words the normalized cosine distance between feature vectors extracted from any pair of coordinates should remain constant between the content image and the output image. This constrains the structure of the output, without enforcing any loss directly connected to pixels of the content image. This causes the semantics and spatial layout to be broadly preserved, while allowing pixel values in $X^{(t)}$ to drastically differ from those in $I_C$.

[1] Since we consider all features to have equal mass, this is a simplified version of the more general EMD [23], which allows for transport between general, non-uniform mass distributions.

### 2.4 User Control

We incorporate user control as constraints on the style of the output. Namely the user defines paired sets of spatial locations (regions) in $X^{(t)}$ and $I_S$ that must have low style loss. In the case of point-to-point user guidance each set contains only a single spatial location (defined by a click). Let us denote paired sets of spatial locations in the output and style image as $(X_{t1}, S_{s1})...(X_{tK}, S_{sK})$. We redefine the ground metric of the Relaxed EMD as follows:

$$C_{ij} = \begin{cases} \beta * D_{cos}(A_i, B_j), \text{ if } i \in X_{tk}, j \in S_{sk} \\ \infty, \text{ if } \exists k \text{ s.t. } i \in X_{tk}, j \notin S_{sk} \\ D_{cos}(A_i, B_j) \text{ otherwise,} \end{cases} \tag{12}$$

where $\beta$ controls the weight of user-specified constraints relative to the unconstrained portion of the style loss, we use $\beta = 5$ in all experiments. In the case of point-to-point constraints we find it useful to augment the constraints specified by the user with 8 additional point-to-point constraints, these are automatically generated and centered around the original to form a uniform 9x9 grid. The horizontal and vertical distance between each point in the grid is set to be 20 pixels for 512x512 outputs, but this is is a tunable parameter that could be incorporated into a user interface.

### 2.5 Implementation Details

We apply our method iteratively at increasing resolutions, halving $\alpha$ each time. We begin with the content and style image scaled to have a long side of 64 pixels. The output at each scale is bilinearly upsampled to twice the resolution and used as initialization for the next scale. By default we stylize at four resolutions, and because we halve $\alpha$ at each resolution our default $\alpha = 16.0$ is set such that $\alpha = 1.0$ at the final resolution.

At the lowest resolution we initialize using the bottom level of a Laplacian pyramid constructed from the content image (high frequency gradients) added to the mean color of the style image. We then decompose the initialized output image into a five level Laplacian pyramid, and use RMSprop [11] to update entries in the pyramid to minimize our objective function. We find that optimizing the Laplacian pyramid, rather than pixels directly, dramatically speeds up convergence. At each scale we make 200 updates using RMSprop, and use a learning rate of 0.002 for all scales except the last, where we reduce it to 0.001.

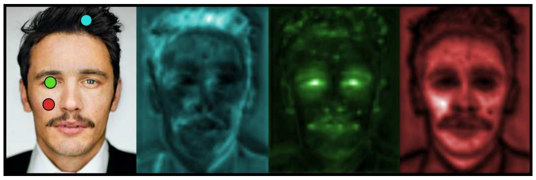


Figure 3: The blue, red, and green heatmaps visualize the cosine similarity in feature space relative to the corresponding points marked in the photograph. Our content loss attempts to maintain the relative pairwise similarities between 1024 randomly chosen locations in the content image

The pairwise distance computation required to calculate the style and content loss precludes extracting features from all coordinates of the input images, instead we sample 1024 coordinates randomly from the style image, and 1024 coordinates in a uniform grid with a random x,y offset from the content image. We only differentiate the loss w.r.t the features extracted from these locations, and resample these locations after each step of RMSprop.

## 3 Related Work

Style transfer algorithms have existed for decades, and traditionally relied on hand-crafted algorithms to render an image in fixed style [7, 9], or hand-crafting features to be matched between an arbitrary style to the content image [10, 3]. The state-of-the-art was dramatically altered in 2016 when Gatys et al. [4] introduced Neural Style Transfer. This method uses features extracted from a neural network pre-trained for image classification. It defines style in terms of the Gram matrix of features extracted from multiple layers, and content as the feature tensors extracted from another set of layers. The style loss is defined as the Frobenius norm of the difference in Gram feature matrices between the output image and style image. The content loss is defined as the Frobenius norm of the difference between feature tensors from the output image and the style image. Distinct from the framework of Neural Style Transfer, there are several recent methods [17, 1] that use similarities between deep neural features to build a correspondence map between the content image and style image, and warp the style image onto the content image. These methods are extremely successful in paired settings, when the contents of the style image and content image are similar, but are not designed for style transfer between arbitrary images (unpaired or texture transfer).

Subsequent work building upon [4] has explored improvements and modifications along many axes. Perhaps the most common form of innovation is in proposals for quantifying the ’stylistic similarity’ between two images [15, 2, 22, 20]. For example in order to capture long-range spatial dependencies Berger et al. [2] propose computing multiple Gram matrices using translated feature tensors (so that the outer product is taken between feature vectors at fixed spatial offsets). Both [4] and [2] discard valuable information about the complete distribution of style features that isn’t captured by Gram matrices.

In [15] Li et al. formulate the style loss as minimizing the energy function of a Markov Random Field over the features extracted from one of the latter layers of a pre-

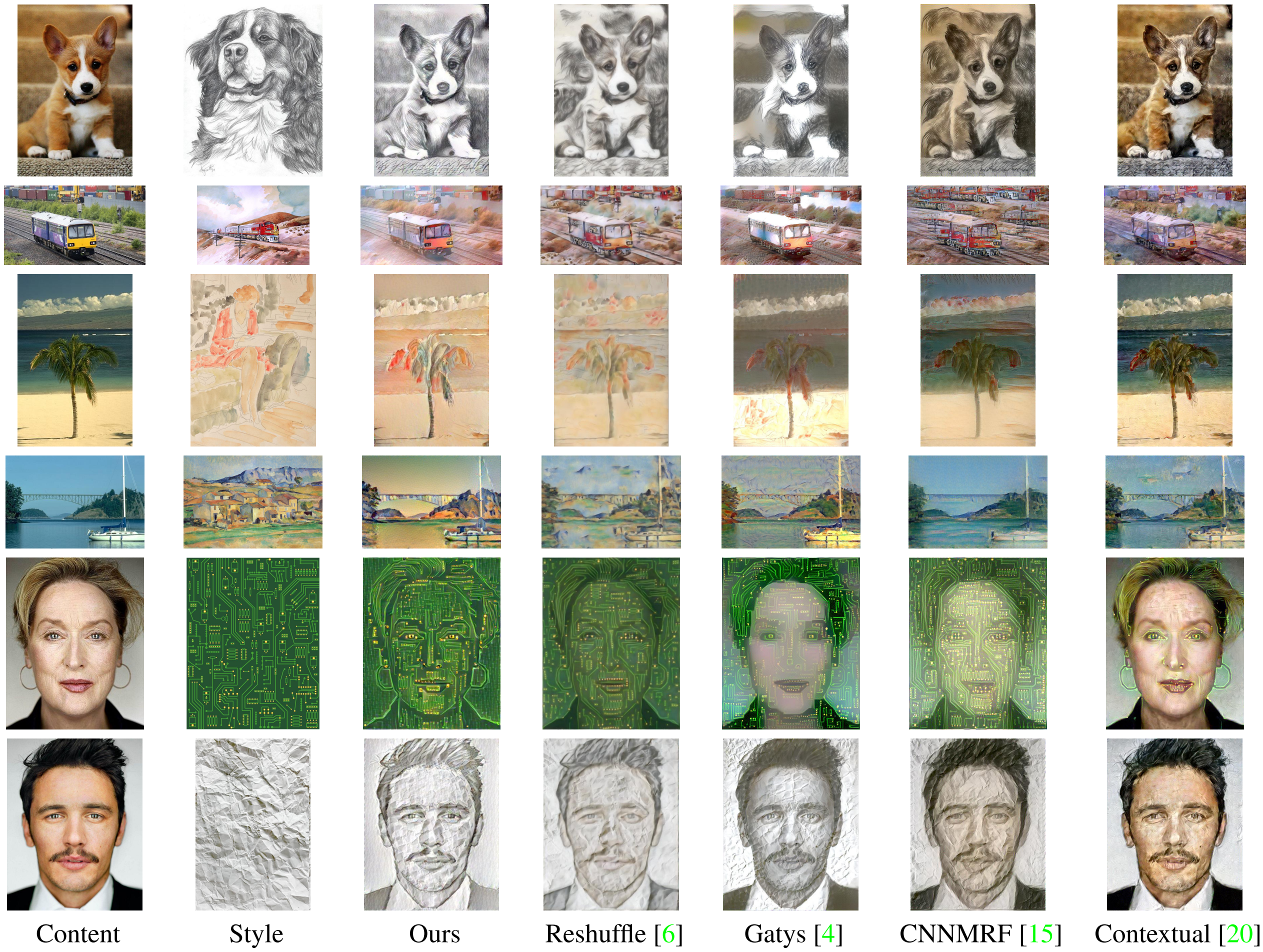


Figure 4: Qualitative comparison between our method and prior work. Default hyper-parameters used for all methods

trained CNN, encouraging patches (which yielded the deep features) in the target image to match their nearest neighbor from style image in feature space. Other functionally similar losses appear in [22], which treats style transfer as matching two histograms of features, and [20], which matches features between the style and target which are significantly closer than any other pairing. In all of these methods, broadly speaking, features of the output are encouraged to lie on the support of the distribution of features extracted from the style image, but need not cover it. These losses are all similar to one component of the Relaxed EMD ($R_A$). However, our method differs from these approaches because our style term also encourages covering the entire distribution of features in the style image ($R_B$). Our style loss is most similar in spirit to that proposed by Gu et al [6], which also includes terms that encourage fidelity and diversity. Their loss minimizes the distance between explicitly paired individual patches, whereas ours minimizes the distance between distributions of features.

Another major category of innovation is replacing the optimization-based algorithm of [4] with a neural network trained to perform style transfer, enabling real-time inference. Initial efforts in this area were constrained to a limited set of pre-selected styles [13], but subsequent work relaxed this constraint and allowed arbitrary styles at test time [12]. Relative to slower optimization-based methods these works made some sacrifices in the quality of the output for speed. However, Sanakoyeu et al. [24] introduce a method for incorporating style images from the same artist into the real-time framework which produces high quality outputs in real-time, but in contrast to our work relies on having access to multiple images with the same style and requires training the style transfer mechanism separately for each new style.

Various methods have been proposed for controlling the output of style transfer. In [5] Gatys et al. propose two 'global' control methods, that affect the entire output rather than a particular spatial region. One method is decomposing the image into hue, saturation, and luminance, and only stylizes the luminance in order to preserve the color palette

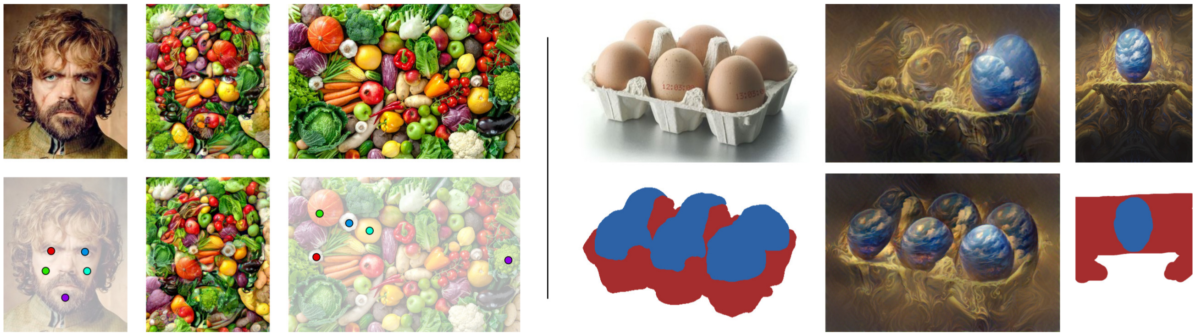


Figure 5: Examples of using guidance for aesthetic effect (left, point-to-point)) and error correction (right, region-to-region). In the top row the images are arranged in order of content, output, style. Below each content and style image we show the guidance mask, and between them the guided output.

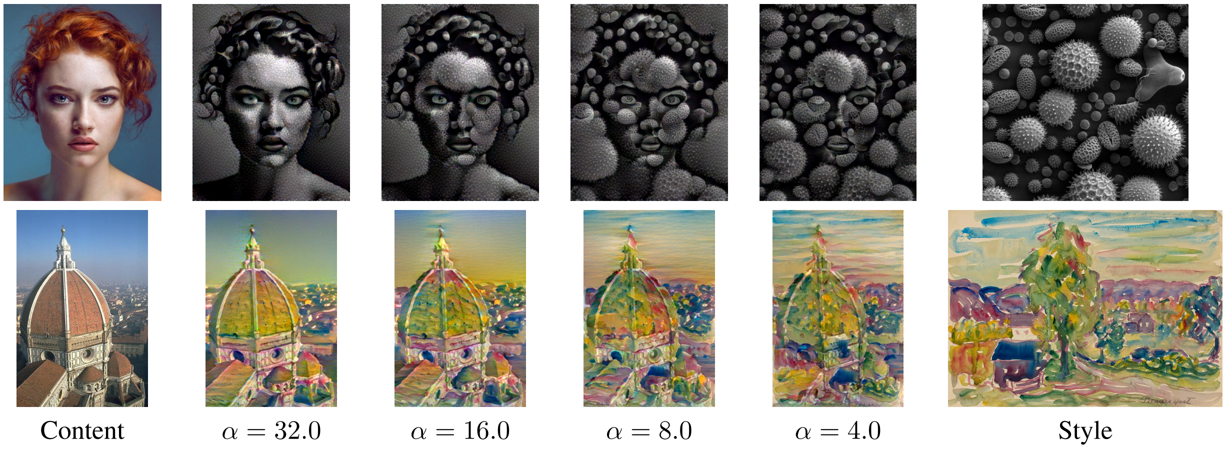
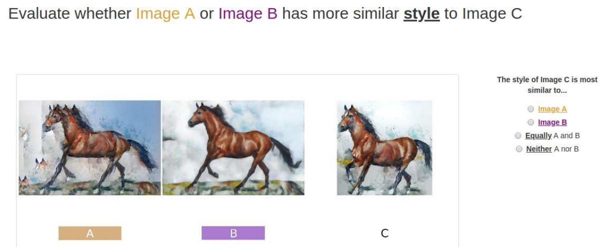


Figure 6: Effect of varying $\alpha$, the content loss weight, on our unconstrained style transfer output, because we stylize at four resolutions, and halve $\alpha$ each time, our default $\alpha = 16.0$ is set such that $\alpha = 1.0$ at the final resolution.

of the content image. A second method from [5] is to generate an auxiliary style image either to preserve color, or to transfer style from only a particular scale (for example the transferring only the brush-strokes, rather than the larger and more structurally complex elements of the style). These types of user control are orthogonal to our method, and can be incorporated into it.

Another type of control is spatial, allowing users to ensure that certain regions of the output should be stylized using only features from a manually selected region of the style image (or that different regions of the output image should be stylized based on different style images). In [5, 18] the authors propose forms of spatial control based on the user defining matched regions of the image by creating a dense mask for both the style and content image. We demonstrate that it is straightforward to incorporate this type of user-control into our formulation of style transfer. In the supplement we show an example comparing the spatial control of our method and [5], and demonstrate that both yield visually pleasing results that match the spatial guidance provided.

Evaluating and comparing style transfer algorithms is a challenging task because, in contrast to object recognition or segmentation, there is no established "ground truth" for the output. The most common method is a qualitative, purely subjective comparison between the output of different algorithms. Some methods also provide more refined qualitative comparisons such as texture synthesis [22, 6] and inpainting [2]. While these comparisons provide insight into the behavior of each algorithm, without quantitative comparisons it is difficult to draw conclusions about the algorithm's performance on average. The most common quantitative evaluation is asking users to rank the output of each algorithm according to aesthetic appeal [6, 16, 19]. Recently Sanakoyeu et al. [24] propose two new forms of quantitative evaluation. The first is testing if an neural network pretrained for artist classification on real paintings can correctly classify the artist of the style image based on an algorithm's output. The second is asking experts in art history which algorithm's output most closely matches the style image. We designed our human evaluation study, described in section 4.1, to give a more complete sense of the trade-off each algorithm makes between content and style as its hyper-parameters vary. To the best of our knowledge it is the first such effort.


Evaluate whether Image A or Image B has more similar **style** to Image C

The style of Image C is most similar to...

○ Image A
○ Image B
○ Equally A and B
○ Neither A nor B

Figure 7: Human evaluation interface

# 4 Experiments

We include representative qualitative results in Figures 2, 4, and an illustration of the effect of the content weight $\alpha$ in Figure 6. Figure 5 demonstrates uses of user guidance with our method.

## 4.1 Large-Scale Human Evaluation

Because style transfer between arbitrary content and style pairs is such a broad task, we propose three regimes that we believe cover the major use cases of style transfer. 'Paired' refers to when the content image and style image are both representations of the same things, this is mostly images of the same category (e.g. both images of dogs), but also includes images of the same entity (e.g. both images of the London skyline). 'Unpaired' refers to when the content and style image are **not** representations of the same thing (e.g. a photograph of a Central American temple, and a painting of a circus). 'Texture' refers to when the content is a photograph of a face, and the style is a homogeneous texture (e.g. a brick wall, flames). For each regime we consider 30 style/content pairings (total of 90).

In order to quantitatively compare our method to prior work we performed several studies using AMT. An example of the workers' interface is shown in Figure 7. Images A and B were the result of the same inputs passed into either the algorithms proposed in [4],[6], [15], [20], or our method. In Figure 7 image C is the corresponding style image, and workers were asked to choose whether the style of image is best matched by: 'A', 'B', 'Both Equally', or 'Neither'. If image C is a content image, workers are posed the same question with respect to content match, instead of style. For each competing algorithm except [6] we test three sets of hyper-parameters, the defaults recommended by the authors, the same with $\frac{1}{4}$ of the content weight (high stylization), and the same with double the content weight (low stylization). Because these modifications to content weight did not alter the behavior of [4] significantly we also tested [4] with $\frac{1}{100}$ and $100\times$ the default content weight. We also test our method with $4\times$ the content weight. We only were able to test the default hyper-parameters for [6] because the code provided by the authors does not expose content weight as a parameter to users. We test all possible pairings of A and B between different algorithms and their hyper-parameters (i.e. we do not compare an algorithm against itself with different hyperparameters, but do compare it to all hyperparameter settings of other algorithms). In each presentation, the order of output (assignment of methods to A or B in the interface) was randomized. Each pairing was voted on by an average of 4.98 different workers (minimum 4, maximum 5), 662 workers in total. On average, 3.7 workers agreed with the majority vote for each pairing. All of the images used in this evaluation will be made available to enable further benchmarking.

For an algorithm/hyper-parameter combination we define its content score to be the number of times it was selected by workers as having closer or equal content to $I_C$ relative to the other output it was shown with, divided by the total number of experiments it appeared in. This is always a fraction between 0 and 1. The style score is defined analogously. We present these results in Figure 8, separated by regime. The score of each point is computed over 1580 pairings on average (including the same pairings being shown to distinct workers, minimum 1410, maximum 1890). Overall for a given level of content score, our method provides a higher style score than prior work.

## 4.2 Ablation Study

In Figure 9 we explore the effect of different terms of our style loss, which is composed of a moment-matching loss $\ell_m$, the Relaxed Earth Movers Distance $\ell_r$, and a color palette matching loss $\ell_p$. As seen in Figure 9, $\ell_m$ alone does a decent job of transferring style, but fails to capture the larger structures of the style image. $\ell_{R_A}$ alone does not make use of the entire distribution of style features, and reconstructs content more poorly than $\ell_r$. $\ell_{R_B}$ alone encourages every style feature to have a nearby output feature, which is too easy to satisfy. Combining $\ell_{R_A}$ and $\ell_{R_B}$ in the relaxed earth movers distance $\ell_r$ results in a higher quality output than either term alone, however because the ground metric used is the cosine distance the magnitude of the features is not constrained, resulting in saturation issues. Combining $\ell_r$ with $\ell_m$ alleviates this, but some issues with the output's palette remain, which are fixed by adding $\ell_p$.

## 4.3 Relaxed EMD Approximation Quality

To measure how well the Relaxed EMD approximates the exact Earth Movers Distance we take each of the 900 possible content/style pairings formed by the 30 content and style images used in our AMT experiments for the unpaired regime. For each pairing we compute the REMD between

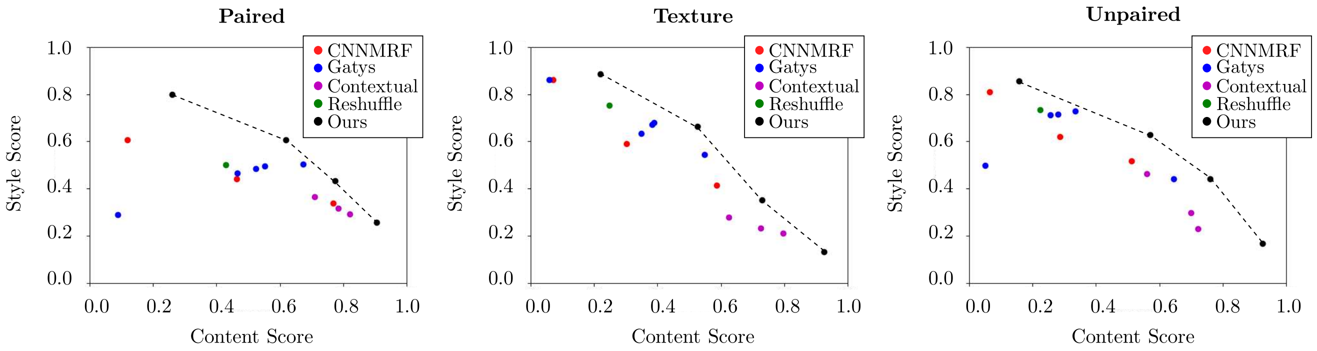
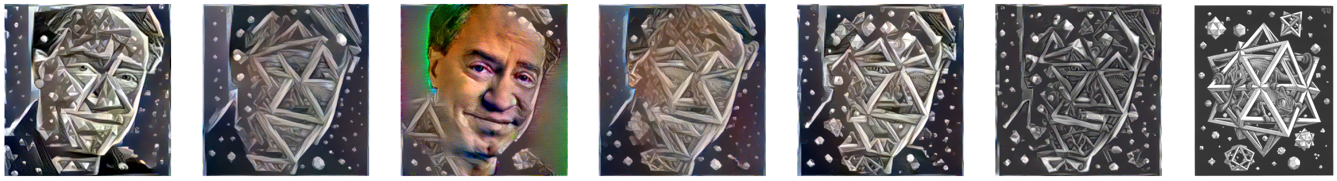


Figure 8: Quantitative evaluation of our method and prior work, we estimate the Pareto frontier of the methods evaluated by linearly interpolation (dashed line)



$\ell_m$ $\ell_{R_A}$ $\ell_{R_B}$ $\ell_r$ $\ell_r + \ell_m$ $\ell_r + \ell_m + \frac{\ell_p}{\alpha}$ Style

Figure 9: Ablation study of effects of our proposed style terms with low content loss ($\alpha = 4.0$). See text for analysis of each terms' effect. Best viewed zoomed-in on screen.

1024 features extracted from random coordinates, and the exact EMD based on the same set of features. We then analyze the distribution of $\frac{REMD(A,B)}{EMD(A,B}$ Because the REMD is a lower bound, this quantity is always $\leq$1. Over the 900 image pairs, its mean was 0.60, with standard deviation 0.04. A better EMD approximation, or one that is an upper bound rather than a lower bound, may yield better style transfer results. On the other hand the REMD is simple to compute, empirically easy to optimize, and yields good results.

### 4.4 Timing Results

We compute our timing results using a Intel i5-7600 CPU @ 3.50GHz CPU, and a NVIDIA GTX 1080 GPU. We use square style and content images scaled to have the edge length indicated in the top row of Table 1. For inputs of size 1024x1024 the methods from [15] and [20] ran out of memory ('X' in the table). Because the code provided by the authors [6] only runs on Windows, we had to run it on a different computer. To approximate the speed of their method on our hardware we project the timing result for 512x512 images reported in their paper based on the relative speedup for [15] between their hardware and ours. For low resolution outputs our method is relatively slow, however it scales better for outputs with resolution 512 and above relative to [15] and [20], but remains slower than [4] and our projected results for [6].

| **Image size** | **64** | **128** | **256** | **512** | **1024** |
|---|---|---|---|---|---|
| Ours | 20 | 38 | 60 | 95 | 154 |
| Gatys | 8 | 10 | 14 | 33 | 116 |
| CNNMRF | 3 | 8 | 27 | 117 | X |
| Contextual | 13 | 40 | 189 | 277 | X |
| Reshuffle | - | - | - | *69** | - |

Table 1: Timing comparison (in seconds) between our methods and others. The style and content images had the same dimensions and were square. *: a projected result, see text for details. -: we were not able to project these results. X: the method ran out of memory.

## 5 Conclusion and Future Work

We propose novel formalizations of style and content for style transfer and show that the resulting algorithm compares favorably to prior work, both in terms of stylization quality and content preservation. Via our ablation study we show that style-similarity losses which more accurately measure the distance between distributions of features leads to better style transfer. The approximation of the earth movers distance that we use is simple, but effective, and we leave it to future work to explore more accurate approximations. Another direction for future work is improving our method's speed by training feed-forward style transfer methods using our proposed objective function.

## References

[1] K. Aberman, J. Liao, M. Shi, D. Lischinski, B. Chen, and D. Cohen-Or. Neural best-buddies. *ACM Transactions on Graphics*, 37(4):114, Jul 2018. 4

[2] G. Berger and R. Memisevic. Incorporating long-range consistency in cnn-based texture generation. *arXiv preprint arXiv:1606.01286*, 2016. 4, 6

[3] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001. 4

[4] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 2, 4, 5, 7, 8

[5] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman. Controlling perceptual factors in neural style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 6

[6] S. Gu, C. Chen, J. Liao, and L. Yuan. Arbitrary style transfer with deep feature reshuffle. 5, 6, 7, 8

[7] P. Haeberli. Paint by numbers: Abstract image representations. In *ACM SIGGRAPH computer graphics*, volume 24, pages 207–214. ACM, 1990. 4

[8] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 447–456, 2015. 2

[9] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 453–460. ACM, 1998. 4

[10] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM, 2001. 4

[11] G. Hinton, N. Srivastava, and K. Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. 2, 4

[12] X. Huang and S. J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. 5

[13] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 5

[14] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015. 1, 3

[15] C. Li and M. 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. 4, 5, 7, 8

[16] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz. A closed-form solution to photorealistic image stylization. *arXiv preprint arXiv:1802.06474*, 2018. 6

[17] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang. Visual attribute transfer through deep image analogy. *SIGGRAPH*, 2017. 4

[18] M. Lu, H. Zhao, A. Yao, F. Xu, Y. Chen, and L. Zhang. Decoder network over lightweight reconstructed feature for fast semantic style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2469–2477, 2017. 6

[19] R. Mechrez, E. Shechtman, and L. Zelnik-Manor. Photorealistic style transfer with screened poisson equation. *arXiv preprint arXiv:1709.09828*, 2017. 6

[20] R. Mechrez, I. Talmi, and L. Zelnik-Manor. The contextual loss for image transformation with non-aligned data. *arXiv preprint arXiv:1803.02077*, 2018. 4, 5, 7, 8

[21] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3376–3385, 2015. 2

[22] E. Risser, P. Wilmot, and C. Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint arXiv:1701.08893*, 2017. 4, 5, 6

[23] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66. IEEE, 1998. 3

[24] A. Sanakoyeu, D. Kotovenko, S. Lang, and B. Ommer. A style-aware content loss for real-time hd style transfer. 2018. 5, 6

[25] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 3

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[27] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 3