Optimal Transport-Based Transforms for Style Transfer

Samuel Boïté

December 17, 2024

1 Introduction

Given a content-style pair, style transfer synthetises a new image that represents the same content but uses the style's characteristics. In 2019, Mroueh [1] improved the results of Li et al. on Universal Style Transfer [2] by applying an optimal transport map between Gaussians to encoded image features. Can we achieve better performance by using an optimal transport between Gaussian mixtures, as introduced by Delon et Desolneux [3]?

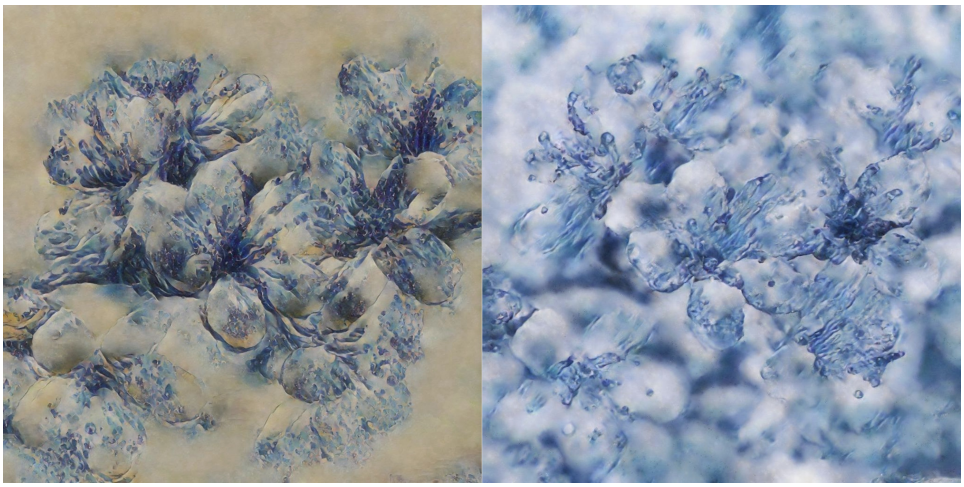

Figure 1.1: Sampled Wasserstein for many content-style pairs [Playground]

2 Background

We place ourselves on $\mathbb{R}^d$, $d \geq 1$. We begin by introducing the 2-Wasserstein distance in the specific context of Gaussian models.

2.1 Wasserstein distance

Definition 1. Let μ_0 and μ_1 be two probability measures in $L^2(\mathbb{R}^d)$. We note $\Pi(\mu_0, \mu_1)$ the set of probability measures γ in $L^2(\mathbb{R}^d \times \mathbb{R}^d)$ such that $P_i \# \gamma = \mu_i$ for $i \in \{0, 1\}$. The 2-Wasserstein distance W_2 between μ_0 and μ_1 is defined as

$$W_2(\mu_0, \mu_1) \coloneqq \left(\inf_{\gamma \in \Pi(\mu_0, \mu_1)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|y_0 - y_1\|^2 \mathrm{d}\gamma(y_0, y_1) \right)^{1/2}.$$

The infimum is attained by some measure $\gamma^\star$ called *optimal transport plan* between μ_0 and μ_1 [4]. Assuming μ_0 and μ_1 are absolutely continuous, we can show that $\gamma^\star$ is unique and has the form

$$\gamma = (\mathrm{Id}, T) \# \mu_0,$$

where $T : \mathbb{R}^d \to \mathbb{R}^d$ is called *optimal transport map* and satisfies $T \# \mu_0 = \mu_1$.

In the Gaussian case, we even have the following closed-form expression:

Proposition 2. Assuming μ_0 and μ_1 are Gaussian, i.e. $\mu_i \sim \mathcal{N}(m_i, \Sigma_i)$, $i \in \{0, 1\}$, and that Σ_is are symmetric semi-definite positive:

$$W_2(\mu_0, \mu_1) = \|m_0 - m_1\|^2 + \mathrm{tr}\left(\Sigma_0 + \Sigma_1 - 2 \left(\Sigma_0^{1/2} \Sigma_1 \Sigma_0^{1/2} \right)^{1/2} \right).$$

Moreover, the optimal transport map can be expressed as

$$T_{\mu_0 \to \mu_1}(x) = m_1 + A(x - m_0),$$

where $A = \Sigma_0^{-\frac{1}{2}} \left(\Sigma_0^{\frac{1}{2}} \Sigma_1 \Sigma_0^{\frac{1}{2}} \right)^{\frac{1}{2}} \Sigma_0^{-\frac{1}{2}} = A^T$.

2.2 Gaussian mixture models

Definition 3. A Gaussian mixture model of size $K \geq 1$ on $\mathbb{R}^d$ is a probability distribution μ on $\mathbb{R}^d$ that can be written

$$\mu = \sum_{k=1}^{K} \pi_k \mu_k, \qquad \text{where } \mu_k \sim \mathcal{N}(m_k, \Sigma_k) \text{ and } \sum_{k=1}^{N} \pi_k = 1.$$

We write $\mathrm{GMM}_d(K)$ the set of Gaussian mixtures on $\mathbb{R}^d$ with less than K components, and $\mathrm{GMM}_d(\infty)$ the set of all finite Gaussian mixtures.

$\mathrm{GMM}_d(\infty)$ is dense in the set of L^2 probability measures for the metric W_2 [4]. We will therefore use Gaussian mixtures to approximate general distributions.

In general, the optimal transport plan $\gamma^\star$ between two Gaussian mixtures μ_0 and μ_1 is not a Gaussian mixture itself. According to [3], this would indeed require the optimal transport map T to be affine, which isn't the case in general. Hence, we cannot ensure that barycenters between Gaussian mixtures remain Gaussian mixtures, leading [3] to introduce the Mixture Wasserstein distance.

2.3 Mixture Wasserstein distance

Let μ_0 and μ_1 be Gaussian mixtures defined by

$$\mu_i \coloneqq \sum_{k=1}^{n} \pi_i^k \mu_i^k, \qquad \mu_i^k \sim \mathcal{N}(m_i^k, \Sigma_i^k), \qquad k \in \{0, 1\}.$$

Definition 4. We define the Mixture Wasserstein distance as the restriction of W_2 over the space of Gaussian mixtures:

$$MW_2^2(\mu_0, \mu_1) \coloneqq \inf_{\gamma \in \Pi(\mu_0, \mu_1) \cap GMM_{2d}(\infty)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|y_0 - y_1\|^2 \mathrm{d}\gamma(y_0, y_1).$$

Then, $(\mathrm{GMM}_d(\infty), MW_2)$ is a geodesic space.

Similarly to $\Pi(\mu_0, \mu_1)$, for π_0 (resp. π_1) in the K_0-dimensional simplex (resp. K_1), we define

$$\Pi(\pi_0, \pi_1) = \left\{ w \in \mathcal{M}_{K_0, K_1}(\mathbb{R}_+), \qquad \forall k, \sum_j w_{kj} = \pi_0^k, \ \forall j \sum_k w_{kj} = \pi_1^j \right\}.$$

Proposition 5. *We can also express the Mixture Wasserstein distance as a discrete problem on the pairwise distances between Gaussian marginals:*

$$MW_2(\mu_0, \mu_1) = \min_{w \in \Pi(\pi_0, \pi_1)} \sum_{k,l} w_{kl} W_2^2(\mu_0^k, \mu_1^l). \tag{2.1}$$

Hence, to compute the Mixture Wasserstein distance, we only need to solve an earth mover problem of size $K_0 \times K_1$, using the matrix of pairwise distances between Gaussian marginals as a cost. From now on, we note $w^\star$ the solution of this discrete minimization problem.

2.4 Transport map for MW_2

To perform style transfer, we need not only an optimal transport plan, but also a transport map giving for each $x \in \mathbb{R}^d$ a corresponding destination $T(x) \in \mathbb{R}^d$. We will use the definition introduced by [3].

Given μ_0 and μ_1 two GMM, the optimal transport plan between them is given by

$$\gamma(x, y) = \sum_{k,l} w^\star_{kl} g_{m_0^k, \Sigma_0^k}(x) \delta_{y=T_{kl}(x)},$$

where $g_{m,\Sigma}$ is the probability density function of a $\mathcal{N}(m, \Sigma)$. It is not of the form $(\mathrm{Id}, T)\#\mu_0$, but we can for instance assign each $x \in \mathbb{R}^d$ to

$$T_{\text{mean}}(x) \coloneqq \mathbb{E}_\gamma(Y|X = x) = \frac{\sum_{k,l} w^\star_{kl} g_{m_0^k, \Sigma_0^k}(x) T_{kl}(x)}{\sum_k \pi_0^k g_{m_0^k, \Sigma_0^k}(x)}. \tag{2.2}$$

3 Experimental Results

As in the Universal Style Transfer method introduced in [2], we will perform style transfer as an image reconstruction process coupled with feature transformation. The reconstruction part is responsable for decoding features back to the RGB space, and the feature transformation translates the content features to the style features.

The full code for these experiments is available on [5].

3.1 Encoder-decoder architecture

We use a pre-trained auto-encoder network for general image reconstruction. It uses VGG-19 [6] as the encoder, and the decoder was trained to invert VGG features back to the original image, using a symmetric architecture. The models' skeleton and checkpoints were taken from [7].

To do a coarse-to-fine stylization, we truncate the VGG encoder up to a specific layer $\text{ReLU}_{X,1}$ ($X \in \{1, \ldots, 5\}$) which empirically carries increasingly low-level information. Using features at all five layers is a way of capturing the characteristics of a style from low levels (e.g. colors) to high levels (e.g. local structures). The full architecture is represented in Figure 3.1.

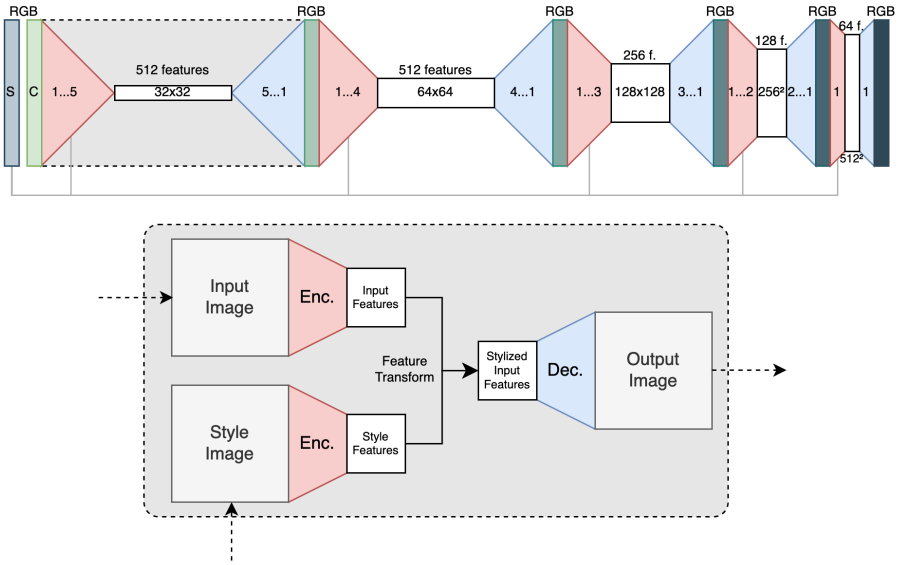

Figure 3.1: VGG-19-based architecture for style transfer

3.2 Feature transforms

We note $(\mathbf{E}, \mathbf{D})$ an encoder-decoder pair and I_c, $I_s \in [0,1]^{3\times(H\times W)}$, we compute the content features $\nu_c = \mathbf{E}(I_c)$ and $\nu_s = \mathbf{E}(I_s)$. Given a specific style transfer method $\mathbf{T}$ and an intensity parameter $\alpha \in [0,1]$, we compute:

$$\nu_\alpha = (1-\alpha)\nu_c + \alpha\mathbf{T}(\nu_c, \nu_s),$$

and we output the image $\tilde{I}^\alpha_{c\to s} = \mathbf{D}(\nu_\alpha)$. We now define several methods $\mathbf{T}$ that we will compare afterwards.

3.2.1 Gaussian Method.

As done by Mroueh [1], using the closed-form expression of Proposition 2, we can define

$$\mathbf{T}(\nu_c, \nu_s) = T_{\hat{\nu}_c, \hat{\nu}_s}$$

where $\hat{\nu}_c$, $\hat{\nu}_s$ are the empirical Gaussian approximations of the former laws. We also define the *Whitening Coloring Transform* (formula in [2]) that, according to [1], is only optimal in the sense of OT when the covariances commute, and the *Mean Transform*, which only readjusts the empirical means ($A = I_d$ in Proposition 2).

3.2.2 Mixture Gaussian Method.

Likewise, we can also define, using the expression of Equation 2.2,

$$\mathbf{T}(\nu_c, \nu_s) = T_{\text{mean}}\left(\hat{\nu}_c^{\text{EM}}, \hat{\nu}_s^{\text{EM}}\right)$$

where $\hat{\nu}_c^{\text{EM}}$, $\hat{\nu}_s^{\text{EM}}$ are Gaussian Mixtures approximating content and style features, obtained using the Expectation-Maximization algorithm [**?**]. We use the *Python Optimal Transport* library [8] for computing T_{mean}, modified to manipulate logarithms of densities only and avoid fractions of very small numbers in Equation 2.2.

3.2.3 Sampled Wasserstein Method.

By sampling $N \approx 10^4$ points of the feature space, a regular computer can find the exact optimal transport plan T for these points, e.g. using POT [8]. We can therefore extend T for every point x in the feature space, by setting $T(x) \coloneqq x + (T(y) - y)$, where y is the closest point to x that was sampled.

3.3 Experiments

3.3.1 Style transfer

For our tests, we use 512×512 images of the Eiffel Tower and Hokusai's painting (see Figure 3.2), and we apply the transforms we've described before, for $\alpha \in \{0.2, 0.4, 0.6, 0.8\}$.

Figure 3.2: Example content-style pair

The results are represented in Figure 3.3: in ascending order, each row corresponds to $\alpha = 0.2$, 0.4, 0.6, 0.8 and columns correspond (from left to right) to Gaussian, GMM(2), GMM(5) and Sampled Wasserstein.

Figure 3.3: Comparison of Gaussian, GMM(2), GMM(5) and Sampled

We note that GMM(2) is a bit more precise than Gaussian, especially when looking at the sky whose color is uniformly more white. However, noise appears as soon as computing GMM(5). This is not due to the irregularity of the transport map itself, as the exact sampled method (last row) yields more regular results. However, when solving Problem 2.1, the pairwise W_2 distances between Gaussian components require computing the inverse square roots of ill-conditioned empirical covariances. Adding a regularization term to the covariance could make them positive, but necessarily at the expense of precision. A solution, still to be explored, could be to threshold the covariances' eigenvalues and perform the transfer in the corresponding subspace only.

3.3.2 GMM barycenters

The Mixture Wasserstein method introduced in Section 3.2.2 gives a natural way to interpolate a given content with many styles. For each style image $i \in \{1, \dots, S\}$, we compute a GMM approximation $\hat{\mu}_{1,i}^{\mathrm{EM}}$ of its encoded features, and we compute a weighted sum of all the GMMs:

$$\hat{\mu}_1 = \sum w_i \hat{\mu}_{1,i}^{\mathrm{EM}}, \qquad \text{where } \sum_{i=1}^{S} w_i = 1.$$

We then apply the Mixture Wasserstein method between content $\hat{\mu}_0^{\mathrm{EM}}$ and aggregated styles $\hat{\mu}_1$ and plot the results in Figure 3.4.

Figure 3.4: Style barycenters using Gaussian mixtures.

4 Conclusion

To conclude, while GMM-based optimal transport (GMM-OT) demonstrated its value for tasks like color transfer—where its regularity proved crucial—the Sampled Wasserstein method ultimately delivered superior visual results for style transfer. Sampling here is only thought as a means to fasten the computing time on a regular laptop. With infinite computing power, one could even apply the exact optimal transport map, which seems to be sufficiently regular. In Figure 1.1, we applied Sampled Wasserstein on two examples.

Nevertheless, GMM-OT retains its usefulness, particularly for tasks like barycenter computation, where it provides a natural method to interpolating between multiple styles. Future work could address the numerical instability issues in this naive implementation of GMM-OT, and try to get less noisy results.

References

[1] Youssef Mroueh. Wasserstein Style Transfer. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 842–852. PMLR, June 2020. ISSN: 2640-3498.

[2] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal Style Transfer via Feature Transforms, November 2017. arXiv:1705.08086.

[3] Julie Delon and Agnes Desolneux. A Wasserstein-type distance in the space of Gaussian Mixture Models, June 2020. arXiv:1907.05254.

[4] Cedric Villani. *Optimal Transport*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, Heidelberg, 2009.

[5] GitHub - samuelbx/deep-style-transfer: Style transfer using VGG19 and optimal transport.

[6] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, April 2015. arXiv:1409.1556.

[7] GitHub - pietrocarbo/deep-transfer: PyTorch implementation of "Universal Style Transfer via Feature Trasforms".

[8] Flamary et al. POT: Python Optimal Transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.