Deep Optimal Transport: A Practical Algorithm for Photo-realistic Image Restoration - Supplementary Material

A Background and extensions

A.1 Numerical Example

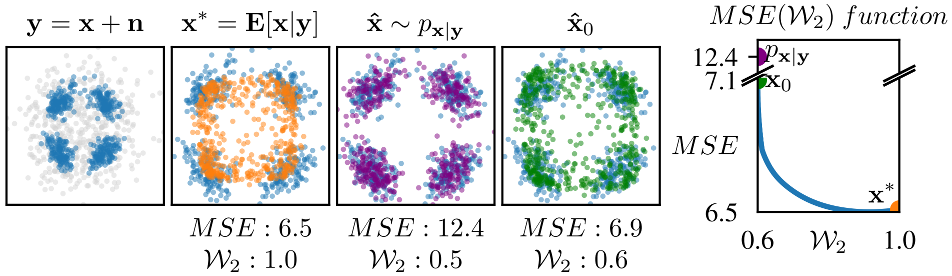

Figure 7: 2D Gaussian mixture denoising. Source samples are shown in blue. The $MMSE$ estimator ($\mathbf{x}^*$, orange) attains the best MSE but the worst perceptual index $\mathcal{W}_2$. The posterior samples ($\mathbf{x}|\mathbf{y}$, purple) attain the best perceptual index but half of the optimal MSE performance. The $\mathbf{D_{max}}$ estimator ($\hat{\mathbf{x}}_0$, green) maintains the MSE of $\mathbf{x}^*$ while attaining a perceptual quality close to $\mathbf{x}|\mathbf{y}$. The DP curve is obtained by interpolating $\hat{\mathbf{x}}_0$ and $\mathbf{x}^*$ using eq. (4).

To guide the reader in understanding the MMSE transport paradigm, we showcase our method on a 2-dimensional denoising problem. To avoid a too trivial uni-modal example, we draw the clean signal from a 4-components Gaussian mixture with non-trivial covariances. We derive linear MMSE and posterior estimators from [38] and proceed by applying the closed-form transport operator introduced in eq. (3).

Note that to avoid deviating from our actual method, we refrain from using more advanced transport operators better suited for multi-modal data. Indeed, those are not a practical solution for real-world image datasets, as they require much more samples than actually available.

We summarize the experiment results in fig. 7. We observe that we obtain the best perceptual quality by sampling from the posterior distribution. However, we witness a significant decrease in MSE performance as predicted by [2]. In contrast, the $\mathbf{D_{max}}$ estimator enjoys a good perceptual index while maintaining a close-to-optimal distortion performance.

A.2 Stochastic transport operator

Throughout our experiments, we found out that increasing the patch-size p can result in numerical instabilities. Recall that the linear transport operator presented in eq. (3) uses the inverse square root of the source covariance matrix $\Sigma_{\mathbf{x}_1}$. When p is large, (typically $p \geq 7$), we obtain ill-conditioned covariance matrices. When the smallest singular value is still positive, we add a small stability constant to the matrix diagonal to ensure it is strictly positive definite. However, the numerical errors sometimes adds up to negative eigenvalues. [1] In this case, we clamp the negative eigenvalues to zero and use the stochastic (one-to-many) transport operator proposed by [1],

$$\mathrm{T}^{\text{stochastic}}_{p_{\mathbf{x}_1} \longrightarrow p_{\mathbf{x}_2}}(x_1) = \Sigma_{\mathbf{x}_2}^{\frac{1}{2}} \left(\Sigma_{\mathbf{x}_2}^{\frac{1}{2}} \Sigma_{\mathbf{x}_1} \Sigma_{\mathbf{x}_2}^{\frac{1}{2}} \right)^{\frac{1}{2}} \Sigma_{\mathbf{x}_2}^{-\frac{1}{2}} \Sigma_{\mathbf{x}_1}^{\dagger} (x_1 - \mu_{\mathbf{x}_1}) + \mu_{\mathbf{x}_2} + w, \tag{5}$$

when $\Sigma_{\mathbf{x}_1}^{\dagger}$ denotes the pseudo-inverse of $\Sigma_{\mathbf{x}_1}$ (after negative eigenvalues where clamped) and $w \sim \mathcal{N}(0, \Sigma_{\mathbf{x}_2}^{\frac{1}{2}}(I - \Sigma_{\mathbf{x}_2}^{\frac{1}{2}} \mathrm{T}^* \Sigma_{\mathbf{x}_1}^{\dagger} \mathrm{T}^* \Sigma_{\mathbf{x}_2}^{\frac{1}{2}})^{\frac{1}{2}} \Sigma_{\mathbf{x}_2}^{\frac{1}{2}})$, with $\mathrm{T}^* = \Sigma_{\mathbf{x}_2}^{-\frac{1}{2}} \left(\Sigma_{\mathbf{x}_2}^{\frac{1}{2}} \Sigma_{\mathbf{x}_1} \Sigma_{\mathbf{x}_2}^{\frac{1}{2}} \right)^{\frac{1}{2}} \Sigma_{\mathbf{x}_2}^{-\frac{1}{2}}$.

---

[1] We tried to avoid overflow when summing over the images by using 64 bit precision

B Practical choices and considerations in our algorithm

B.1 Working in latent space

We adopt the latent transport approach where the images are embedded into the latent space of a pre-trained auto-encoder. Let $\mathbf{E}(\cdot)$, $\mathbf{D}(\cdot)$ denote the encoder and decoder, respectively. Even if $\mathbf{D}(\mathbf{E}(t)) = t$, it is likely that $\mathbf{E}(\cdot)$ "deforms" the space, I.e., $\|\mathbf{E}(s) - \mathbf{E}(t)\| \neq \|s - t\|$, which means that the optimal transport plan in the latent space could be *different* than the plan we seek in the pixel space (the cost function in eq. (3) has changed). We can address this by modifying the latent cost function to account for the deformation via the following change of variables

$$\mathbb{E}\left[\|\hat{\mathbf{x}} - \mathbf{x}\|^2\right] = \mathbb{E}\left[\frac{\|\mathbf{E}(\hat{\mathbf{x}}) - \mathbf{E}(\mathbf{x})\|^2}{|\frac{\partial \mathbf{E}(\mathbf{x})}{\partial \mathbf{x}}| \cdot |\frac{\partial \mathbf{E}(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}}|}\right], \tag{6}$$

where $|\frac{\partial \mathbf{E}(\mathbf{x})}{\partial \mathbf{x}}|$ is the determinant of the Jacobian matrix of $\mathbf{E}(\cdot)$ However it is not a practical solution since we lose access to the closed-form solution eq. (2). Note that the latent MSE approximation is usually desirable when dealing with natural images (e.g. to elaborate image quality measure [39], perceptual quality metrics [25]). It is also true in our case but it means we can no longer claim we obtain the $\mathbf{D_{max}}$ estimator.

With that, we argue that switching to a latent cost is actually a strength rather than a weakness of our method. Indeed, using the MSE between deep latent variables has shown to be a better fit to compare natural images than directly working in the pixel space [32]. The authors of [7] trained their VAE (which is used in our experiments) to remove "imperceptible details" from the latent representation, in order to better focus on higher level image semantics. In section 5.1 we validate this claim by showing that our algorithm maintains the "perceptual" discrepancy performance of the original estimator (*e.g.*, LPIPS).

B.2 Overlapping patches extraction strategy

For Convolutional Neural Network (CNN) encoders [2], let (c, H_e, W_e) denote the shape of the latent representation (CNN encoders produce 3-dimensional encoded tensors), where H_e, W_e the spatial extent and c is the number of channels (i.e., the number of convolution kernels in the last convolution layer). The covariance matrices $\Sigma_{\hat{\mathbf{x}}_e}$, $\Sigma_{\mathbf{x}_e}$ contain $\frac{(c, H_e, W_e)^2}{2}$ parameters, which may require a large amount of samples for large latent images with $H_e, W_e \gg 1$. To mitigate the quadratic dependency on $H_e \cdot W_e$, we assume that the latent pixels depend only on the pixels in their close neighborhood. In practice, we unfold the latent representation, extracting all overlapping patches of shape (c, p, p). A similar approximation exists in the style-transfer literature [8, 9], where instead of patches, only the pixels are considered (i.e., this is a private case of our approach with $p = 1$). In section 5.3 we empirically show that increasing p improves the perceptual quality at the expense of MSE performance, given that enough training samples are available.

B.3 Shared distribution

When dealing with natural image scenes, it is beneficial to suppose that overlapping patches share common statistical attributes [39, 40]. In the case of a CNN encoded image, this approximation remains satisfying because we ultimately look at filter activations which are spatial-invariant with each latent patch having the same receptive field. Therefore, we assume that the overlapping patches are all samples from the same distribution. This approach dramatically reduces the number of estimated parameters, and also multiplies the number of samples at our disposal by $H_e \cdot W_e$, which alleviates the curse of dimensionality. We demonstrate these practical benefits in section 5.3. In practice, given N images, we "flatten" all the extracted patches to vectors $\underline{v}_{cp^2 \times 1}$ which we stack into a sample matrix $\underline{\underline{X}}_{NH_eW_e \times cp^2}$. We then aggregate the samples to compute the MVG statistics: $\mu = X^T \mathbf{1}$, $\Sigma = \frac{NH_eW_e}{NH_eW_e - 1}(X - \mu)(X - \mu)^T$. As NH_eW_e may be very large, we perform all computations in double precision. When training, this process is done twice; once for the natural image samples, and once for the restored samples we wish to transport.

---

[2] This methodology can easily be extrapolated to other encoder architectures.

B.4 Size of the latent representation

When increasing the capacity of models with a fixed encoding rate, deepening is preferable than widening. Indeed, increasing c makes the covariance estimation dramatically harder while increasing H_e, W_e enlarges the sample pool. Therefore, the VAE from [7] with $c = 4$ and $H_e, W_e \gg 1$ is a particularly good candidate for our method. For $p = 3$ for instance, the covariance matrix admits only 1296 parameters while each 512^2 image contributes 4096 samples to its estimation. As we see next, this greatly contributes to reducing the number of training samples needed to estimate the covariance matrices and allows to compute the transport operator in a few-shot manner.

B.5 Transport

In a single pass on a data set of natural images and a (possibly different) data set of restored samples, we compute $\mathrm{T}^{\mathrm{MVG}}_{p_{\hat{\mathbf{x}}_e} \longrightarrow p_{\mathbf{x}_e}}$ (see eq. (2)). Note that each latent distribution could sometimes be degenerate, especially for severe degradations. Fortunately, the classical MVG transport operator can be generalized to ill-posed settings where $\Sigma_{\hat{\mathbf{x}}}$ is a singular matrix (see appendix A.2).

B.6 Decoding

Since the transported patches overlap, we "fold" them back into a latent image $\hat{\mathbf{x}}_{0,latent}$ by averaging. The latent image is then decoded back to the pixel space, i.e. $\hat{\mathbf{x}}_0 = \mathbf{D}(\hat{\mathbf{x}}_{0,latent})$. Since $\mathbf{E}(\cdot)$ is not invertible, the decoder $\mathbf{D}(\cdot)$ is used as a convenient approximation in the training domain of the auto-encoder. A corollary of this approximation is that the auto-encoder should in theory be trained on the image distribution we aim to transport, which weakens our claim to a fully blind algorithm.

All the steps described above are summarized in fig. 4.

B.7 Transporting the degraded measurement

We tried applying our algorithm on the degraded measurement directly. Indeed we observe qualitatively and quantitatively that transporting the degraded measurement $\mathbf{y}$ amplifies the degradation (refer to fig. 8).

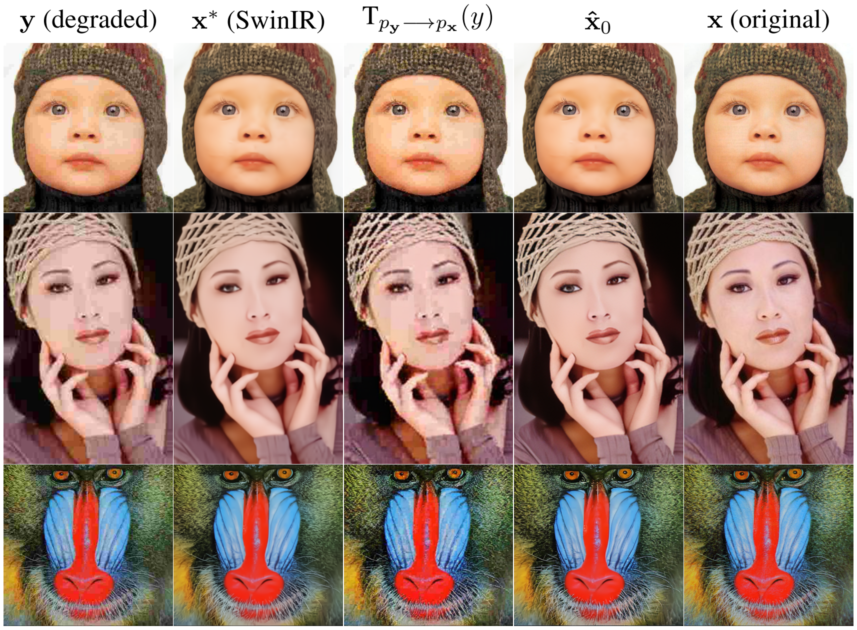

Figure 8: Transporting the degraded measurement (JPEG$_{q=10}$) directly is not enough to restore the image. It can sometimes even exacerbate the degradation. Quantitatively, the degraded sample $\mathbf{y}$ has better PSNR and FID than its transported version (respectively 27.26 dB and 13.88 FID *v.s.* 23.69 dB and 15.88 FID).