Generative Modeling Report on Project 5
Pseudoinverse-Guided Diffusion Models for Inverse Problems

Léa Bohbot and Jérémie Touati

1 Introduction and context

1.1 Diffusion models

The article under scrutiny falls within the framework of diffusion models. Their initial goal is to model a probability distribution in a space of images and generate samples from this distribution. In order to do so, noise is progressively added to an image, which is equivalent to carrying it from the space of images to the space of noise. The diffusion models are trained to learn how to denoise this image step by step, that is, to bring it back to the image distribution space.

In continuous diffusion models using stochastic differential equations (SDE), the first step of progressively adding noise is completed through equation (1), written here in the simplified case of Ornstein-Uhlenbeck. The backward process follows equation (2).[1]

$$\text{Forward (Ornstein-Uhlenbeck):} \quad \mathrm{d}x_t \;=\; -\,x_t\,\mathrm{d}t \;+\; \sqrt{2}\,\mathrm{d}B_t \tag{1}$$

$$\text{Backward:} \quad \mathrm{d}x_{T-t} \;=\; \big(x_{T-t} \;+\; 2\,\nabla \log p_{T-t}(x_{T-t})\big)\,\mathrm{d}t \;+\; \sqrt{2}\,\mathrm{d}B_t \tag{2}$$

where $\nabla \log p_{T-t}(x_{T-t})$ is the score function, that allows to inverse the diffusion process induced by the stochastic term $\sqrt{2}\,\mathrm{d}B_t$ in the forward process. It is learned through a neural network.

1.2 The DDPM framework

Here we study a discrete version of this same problem called a DDPM (Denoising Diffusion Probabilistic Model). We consider a fixed number of transitions steps, and the processes becomes

$$\text{Forward:} \quad p_\theta(x_{0:T}) \;=\; p(x_T)\prod_{t=1}^{T} p_\theta\big(x_{t-1} \mid x_t\big), \quad \text{where} \quad p_\theta\big(x_{t-1} \mid x_t\big) \;=\; \mathcal{N}\big(\mu_\theta(x_t,t),\, \beta_t\, I_d\big)$$

$$\text{Backward:} \quad q(x_{0:T}) \;=\; q(x_0)\prod_{t=1}^{T} q\big(x_t \mid x_{t-1}\big), \quad \text{where} \quad q\big(x_t \mid x_{t-1}\big) \;=\; \mathcal{N}\big(\sqrt{1-\beta_t}\,x_{t-1},\, \beta_t\, I_d\big)$$

with the simplification assumption that p_θ is gaussian. Here we consider the diffusion as a fixed stochastic encoder where

$$x_t \;=\; \sqrt{1-\beta_t}\,x_{t-1} \;+\; \sqrt{\beta_t}\,z_t, \quad \text{with} \quad \beta_t \in (0,1), \quad \varepsilon \sim \mathcal{N}(0,I), \quad \varepsilon \perp\!\!\!\perp x_0$$

which can be reformulated by

$$x_t \;=\; \sqrt{\bar{\alpha}_t}\,x_0 \;+\; \sqrt{1-\bar{\alpha}_t}\,\varepsilon, \quad \text{with} \quad \alpha_t = 1-\beta_t \quad \text{and} \quad \bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$$

According to the principle of variational inference (not detailed here), we train the network by maximizing the following ELBO:

$$\mathbb{E}\big[-\log p_\theta(x_0)\big] \;\;\leq\;\; \mathbb{E}_q\Big[-\log\Big(\frac{p_\theta(x_{0:T})}{q\big(x_{1:T} \mid x_0\big)}\Big)\Big] \;:=\; L.$$

[1] NB: we follow the notations from the course, and we will detail their correspondence with the notations of the article.

Expliciting this ELBO leads to the computation of the KL divergence $D_{\mathrm{KL}}\big(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t)\big)$. One can show that

$$q\big(x_{t-1} \mid x_t, x_0\big) \sim \mathcal{N}\Big(\tilde{\mu}(x_t, x_0),\, \tilde{\beta}_t\, I_d\Big)$$

$$\text{with} \quad \tilde{\mu}(x_t, x_0) \;=\; \frac{\sqrt{\bar{\alpha}_{t-1}}\,\beta_t}{1-\bar{\alpha}_t}\, x_0 \;+\; \frac{\sqrt{\alpha_t\,(1-\bar{\alpha}_{t-1})}}{1-\bar{\alpha}_t}\, x_t, \quad \tilde{\beta}_t \;=\; \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\,\beta_t$$

For two gaussian distributions, $D_{\mathrm{KL}}\big(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t)\big) \;=\; \frac{1}{\beta_t}\left\|\mu_\theta(x_t, t) - \tilde{\mu}(x_t, x_0)\right\|^2 + C$. So we focus here on learning the drift μ_θ.

We can either solve this by rewriting everything as a function of the added standard noise ϵ, like in the course and in TP6. We can also rewrite it using the score function. The latter approach is the one chosen by Chung et al. [1] and the authors of the paper we studied [2]. In section 2.2 we use Tweedie formula to show that those two formulations are equivalent.

Finally we have

$$\mu_\theta(x_t, t) \;=\; \frac{1}{\sqrt{\alpha_t}}\Big(x_t \;-\; \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(x_t, t)\Big),$$

$$L_t \;=\; \frac{\beta_t}{1-\bar{\alpha}_t}\,\mathbb{E}_q\big[\|\epsilon_\theta(x_t, t) \;-\; \epsilon\|^2\big] \;+\; C.$$

The noise added to x_0 is predicted using a U-Net, and a gradient descent is performed on the loss L_t.

1.3 Link to inverse problems

This framework of diffusion models is thus very general and not spectific to a task. Here we are interested in applying them to solve inverse problems. We want to recover some signal $\mathbf{x}_0 \in \mathbb{R}^n$ from a measurement $\mathbf{y} \in \mathbb{R}^m$ such that

$$\mathbf{y} = H\mathbf{x}_0 + \mathbf{z},$$

where $H \in \mathbb{R}^{n \times m}$ is the known measurement matrix (model), and $\mathbf{z} \sim \mathcal{N}(0, \sigma_v^2 \mathbf{I})$ is an i.i.d. noise vector with known variance.

In order to do so, we can condition the sampling to our particular task by adding problem-specific scores $\nabla_{x_t} \log p_t\big(x_t \mid \mathbf{y}\big)$. The traditional score is already a guidance term that is supposed to direct the reverse diffusion process towards the right distribution. The idea here is to add a guidance which is conditioned on a task to bias the generation towards this particular goal/inverse problem. In other words, we condition our generation on the measurement y in order to guide the model to reconstruct this particular measurement (and not some random image).

At this point, we can either train a problem-agnostic model or specialize our conditioned model for a particular H. The latter option can be more costly as it requires to train a model for every task to solve. The issue with problem-agnostic model is that their generality comes with a cost: they often perform worse. This is the problem that conditional diffusion models for inverse problems tackle, and what both Chung and Song ponder on [2]. The choice of the conditional guidance is the main contribution of this article.

2 Theoretical comparison with Chung et al.

2.1 Comparing the scores approximations

Both authors tackle the issue of the approximation of the problem specific score:

$$\nabla_{x_t} \log p_t\big(x_t \mid \mathbf{y}\big) \;=\; \nabla_{x_t} \log p_t\big(x_t\big) \;+\; \nabla_{x_t} \log p_t\big(\mathbf{y} \mid x_t\big)$$

The first term is the traditional score approximated with the score network $S_\theta(x_t;\sigma_t)$, or deduced with Tweedie's formula. And the second term is the score of:

$$p_t\big(\mathbf{y} \mid x_t\big) \;=\; \int_{x_0} p_t\big(x_0 \mid x_t\big)\; p\big(\mathbf{y} \mid x_0\big)\, \mathrm{d}x_0$$

This probability is intractable. Indeed, while $p\big(\mathbf{y} \mid x_0\big)$ can be computed, sampling from $p_t\big(x_0 \mid x_t\big)$ is equivalent to sampling from the whole diffusion model which is infeasible, even using Monte-Carlo methods.

The approach of Chung et al. consists in simplifying the expectation over $p_t\big(x_0 \mid x_t\big)$ by taking

$$p_t\big(\mathbf{y} \mid x_t\big) \approx p\big(\mathbf{y} \mid x_0\big)$$

As $\mathbf{y} = H\mathbf{x}_0 + \mathbf{z}$ we can easily derive

$$p(\mathbf{y} \mid x_0) \;=\; \frac{1}{\sqrt{(2\pi)^n\,\sigma^{2n}}}\, \exp\Big(-\frac{\|\mathbf{y} - Hx_0\|_2^2}{2\,\sigma^2}\Big)$$

The last step is to approximate $x_0 \approx \hat{x}_0(x_t)$, which gives the score

$$\nabla_{x_t}\, \log\, p(\mathbf{y} \mid x_t) \;\simeq\; -\,\frac{1}{\sigma^2}\, \nabla_{x_t}\, \big\|\mathbf{y} - H\hat{x}_0(x_t)\big\|_2^2$$

The choice made by Song et al. is less simplistic. They approximate $p_t\big(x_0 \mid x_t\big)$ by a Gaussian, using only the diffusion process. As explained in the article this is equivalent to representing $p_t\big(x_0 \mid x_t\big)$ by using a one-step denoising, instead of the entire diffusion model from time t to 0. In other words, $x_0 \mid x_t$ becomes approximately the result of the denoised estimator $\hat{x}_t$ of x_t at time t (with a certain variance). More precisely

$$p_t\big(x_0 \mid x_t\big) \;\approx\; \mathcal{N}\big(\hat{x}_t,\, r_t^2\, \mathbf{I}\big)$$

And using Tweedie's formula:

$$\hat{x}_t \;=\; \mathbf{x}_t \;+\; \sigma_t^2\, \nabla_{x_t}\, \log\, p_t(\mathbf{x}_t) \;\approx\; \mathbf{x}_t \;+\; \sigma_t^2\, S_\theta\big(\mathbf{x}_t;\sigma_t\big)$$

Then we derive $p_t\big(\mathbf{y} \mid x_t\big)$:

$$p_t\big(\mathbf{y} \mid x_t\big) \;\approx\; \mathcal{N}\Big(H\,\hat{x}_t,\;\; r_t^2\, H\, H^\top \;+\; \sigma_y^2\, \mathbf{I}\Big)$$

Hence, an approximate expression for the score is:

$$\nabla_{x_t}\, \log\, p_t\big(\mathbf{y} \mid x_t\big) \;\approx\; \left(\big(\mathbf{y} - H\,\hat{x}_t\big)^\top \Big(r_t^2\, H\, H^\top + \sigma_y^2\, \mathbf{I}\Big)^{-1}\, H\, \frac{\partial\, \hat{x}_t}{\partial\, x_t}\right)^\top \tag{3}$$

We note that this approximation does not make any assumption about the underlying problem we want to solve, that is H can be linear, non linear, noise can be added or not... In addition, the derivation is only performed through the score model $\hat{x}_t$ and not through the graph of the matrix H. So H does not even need to be differentiable, on contrary to what was assumed in Chung et al.

A simplified version of (3) can be formulated in the noiseless case $\sigma_y = 0$:

$$\nabla_{x_t}\, \log\, p_t\big(\mathbf{y} \mid x_t\big) \;\approx\; r_t^{-2}\, \Big(\big[\, H^\dagger\, \mathbf{y} \;-\; H^\dagger\, H\, \hat{x}_t\big]^\top\, \frac{\partial\, \hat{x}_t}{\partial\, x_t}\Big)^\top$$

where $H^\dagger$ is the Moore–Penrose pseudoinverse defined by $H^\dagger = H^\top\, \big(H\, H^\top\big)^{-1}$. Furthermore, in this noiseless case, Song et al. extend by analogy this formula to any non-linear operator h using:

$$\nabla_{x_t}\, \log\, p_t\big(\mathbf{y} \mid x_t\big) \;\approx\; r_t^{-2}\, \Big(\big[h^\dagger(\mathbf{y}) \;-\; h^\dagger\big(h(\hat{x}_t)\big)\big]^\top\, \frac{\partial\, \hat{x}_t}{\partial\, x_t}\Big)^\top,$$

where $h(\cdot)$ is the measurement model and $h^\dagger(\cdot)$ a pseudoinverse (or approximate inverse).

2.2 Equivalence between score and noise formulations of the problem

The equivalence of those two formulations is derived in the appendix section A.

3 Choice of variance weights and guidance's coefficient

3.1 Choice of the adapted weights for the variance

As we said earlier, the inherent difficulty of computing

$$p(y|x_t) = \int_{x_0} p(x_0|x_t)p(y|x_0)\,dx_0$$

comes from the fact that $p(x_0|x_t)$ is intractable (as it can only be approximated by the all diffusion model with high precision).

Song et al. propose a Gaussian approximation whose variance is supposed to depend on the data and the time t:

$$p(x_0|x_t) \approx \mathcal{N}(x_0; \hat{x}_t, r_t^2)$$

The problem is now to choose a value of r_t that makes sense. Ho et al. [3] propose to take

$$r_t^2 = \sigma_t^2 \tag{4}$$

Song "corrects" this variance by relying on the following reasoning: under the hypothesis that $p(x_0)$ is Gaussian, if we come back to Bayes' rule, we have:

$$p_t(x_0|x_t) \propto p_0(x_0)p_t(x_t|x_0) = \mathcal{N}\left(x_t; \frac{x_t}{\sigma_t^2+1}, \frac{\sigma_t^2}{\sigma_t^2+1}I\right) \tag{5}$$

So by identification, they take:

$$r_t^2 = \frac{\sigma_t^2}{\sigma_t^2+1} \tag{6}$$

We can demonstrate this by using the conjugation of two Gaussian. This derivation of equation (5) is made in the section B of the appendix. Finally, with our notation, we have:

$$\sigma_t^2 = 1-\bar{\alpha}_t, \quad r_t^2 = \frac{1-\bar{\alpha}_t}{2-\bar{\alpha}_t}$$

We noticed that the Ho et al. [3] proposition was more stable for a lot of images in our case, and we make an empirical comparison on an image in the following section.

3.2 Choice for the guidance's coefficient: Song vs Chung

Using the DDPM sampler, determining the appropriate guidance coefficient requires examining the mean expression:

$$\tilde{\mu} = \frac{1}{\sqrt{\alpha_t}}x_t + \frac{\beta_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}\hat{s}$$

where $\hat{s} = \hat{s}_0 + \hat{s}_{\text{guidance}}$ with $\hat{s}_0 = \nabla_x \log p(x_t)$ and $\hat{s}_{\text{guidance}} = \nabla_x \log p(x_t|y)$. This yields:

$$\tilde{\mu} = \underbrace{\left(\frac{1}{\sqrt{\alpha_t}}x_t + \frac{\beta_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}\hat{s}_0\right)}_{\tilde{\mu}_0} + \frac{\beta_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}\hat{s}_{\text{guidance}}$$

Following this derivation, we would scale the conditional guidance $\nabla_{x_t} \log p_t(y|x_t)$ by $\frac{\beta_t}{\sqrt{\alpha_t}(1-\bar{\alpha}_t)}$. However, our tests revealed this approach to be unstable.

Both Song and Chung instead directly add the guidance term to the equation. Chung divides the guidance $\nabla_{x_t}\|y - H\hat{x}\|^2$ by $\xi \times \frac{1}{\|y - H\hat{x}\|}$ where ξ is a task-dependent empirical coefficient.

Knowing that $\nabla_{x_t}\|y - H\hat{x}\|^2 = 2\|y - H\hat{x}\|\nabla_{x_t}\|y - H\hat{x}\|$, we interpret this as a way to preserve only directional information, with the gradient scaled by $\|y - H\hat{x}\|$ plus an adaptive factor.

Song alternatively adds their guidance term directly to $\tilde{\mu}_0$, adapting to VP formulation by rescaling with $\sqrt{\alpha_t}$. Since our formulation is already VP-based, this would require using a coefficient of 1 for the guidance — a method we found entirely unstable.

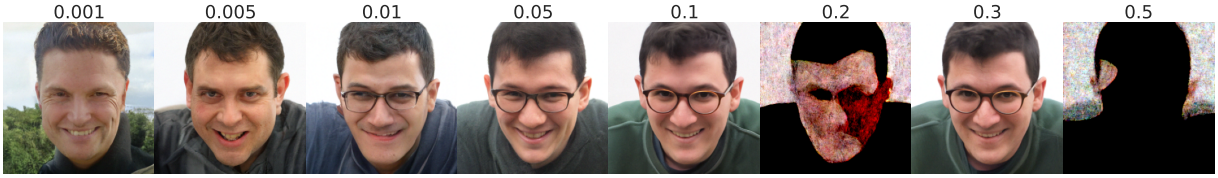
3.3 Our choice for the guidance's coefficient

Our final choice is inspired from the method of Chung and relies on the order of magnitude of the formula. Given our guidance, we choose an adaptive coefficient by taking the square root of the norm of the guidance, scaled by a task-dependent factor zeta:

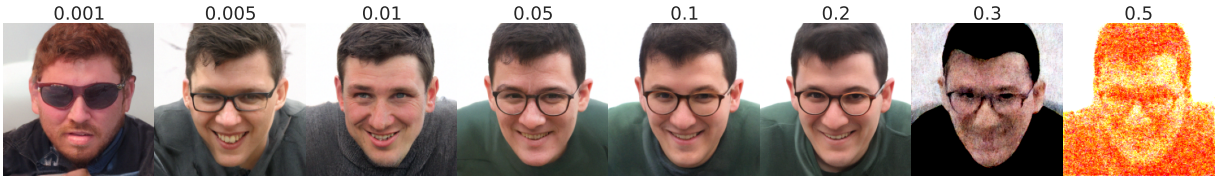
$$\frac{\zeta}{\sqrt{||\mathbf{guidance}||}}$$

In figure 1, we compare Ho's and Song's approaches for the value of r_t^2 (see eq. (4) and (6)) on a simple denoising example with $\sigma^2_{noise} = 0.1$. We test several values of ζ in front of the guidance term, from 0.001 to 0.5.

(a) Original image and noisy version ($\sigma^2_{noise} = 0.1$)



(b) Ho's approach



(c) Song's approach

Figure 1: Pseudo-inverse guidance reconstructions for a simple denoising task. Ho's and Song's approaches are compared for the choice of the variance. Several values of the guidance scaling factor ζ (from 0.001 to 0.5) are tested.

For both cases, a low scaling factor ζ gives a generated face of good quality, but very different from the desired one. The backward process has not been guided enough towards the original image. As the guidance factor increases, the look and posture of the reconstructed person gets closer to the desired one, until reaching an optimal resemblance result for a value of 0.1 in Ho's approach and 0.2 in Song's one. When the guidance becomes too important, the reconstruction seems to explode: it

gets too far from the distribution of images learned by the model.

Compared with their respective optimal guidance factors, Ho's approach turns out to be more satisfying to our perception than Song's one. We notice that it its rendering of the details is better (cf. the collar and the lines on the man's tee-shirt). For the following experiments, we will then stick to Ho's approach. Finally, for each reconstruction task, we will choose a specific guidance factor.

4 Experiments and comparison with Chung et al

For the following experiments, we base ourselves on the DDPM of TP6, itself based on [4] and [1] with a U-Net trained by Chung et al. [1] Of course, we adapt for each inverse problem the guidance term to match the theory of our article. We describe in the following sections how we do this for the tasks of denoising, inpainting, and Gaussian and uniform deblurring.

4.1 Metrics used

For the denoising task, we use the PSNR (peak signal to noise ratio). Regarding the inpainting task, we would like to assess the quality of the inpainted images using the FID (Fréchet Inception Distance). However, this metric compares two sets of images and requires to have thousands of images in both of them. Given the time required to sample a single image, we thus do not consider this metric. For this reason, we decide to use the LPIPS (Learned Perceptual Image Patch Similarity). Instead of comparing two sets of images, the LPIPS compares two given images [5] by computing their activations for some pre-trained neural network (VGG in our case). The similarity between those activations has been shown to match the human perception of resemblance.

4.2 Denoising

The denoising task is the simple inverse problem for which $H = I$. In this case, the guidance term given by equation (3) is straightforward to compute, leveraging the pseudo-code given in the appendix of the article.

We compare Chung's method with the pseudo-inverse guidance on 5 images from the FFHQ dataset [6] with values in $[-1; 1]$. We add Gaussian noise to all of them with 7 levels of noise, ranging from 0.01 to 0.5. For this task, we choose ζ for the guidance term equal to 0.1. Examples of such images with their noisy versions and their reconstructions using both Chung's and Song's method are shown in appendix (figure 3). Figure 2a presents the average and standard deviation of the PSNR over the 5 images for both methods.

In average, the pseudoinverse guidance performs a better denoising (higher PSNR) than Chung. However, the quality of the denoising seems to vary more than with Chung, where the standard deviation is constant and smaller. These fluctuations may also come from the fact that our set of images only contains 5 samples. We were indeed limited by computational constraints, but further work could include performing more robust evaluations.

4.3 Inpainting

Let x be a 3-channel image of size $n \times n$. Let $\delta_{i,j,c} \in \{0, 1\}$ be the variable determining whether or not the channel c of pixel (i, j) is masked (0 corresponds to masking the pixel, 1 to letting it unchanged). Then masking x can be performed by applying the term-wise multiplication

$$y = M * x$$

where M is also a 3-channel image of size $n \times n$ where the value at pixel (i, j) and channel c is given by $\delta_{c,i,j}$.

If we see the image x as a $\mathbb{R}^{3\times n\times n}$ vector, then the masking can be seen as a linear operator in the vector space:

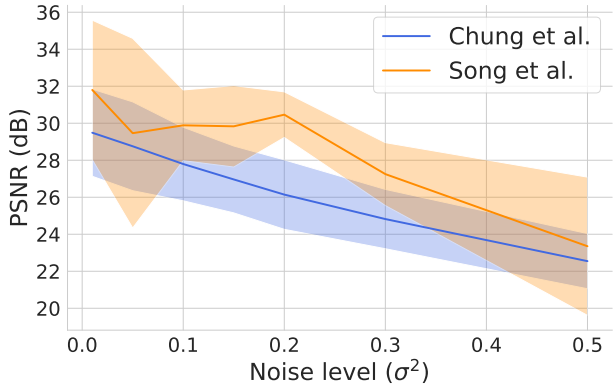
$$H_{mask} = \begin{bmatrix} \delta_{1,1,1} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & \delta_{2,1,1} & 0 & \cdots & \cdots & 0 \\ 0 & 0 & \delta_{3,1,1} & \cdots & \cdots & 0 \\ 0 & 0 & 0 & \delta_{1,1,2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \delta_{3,n,n} \end{bmatrix} = \mathrm{Diag}\big(\delta_{c,i,j}\big)_{c,i,j} \in \mathbb{R}^{(3\times n\times n)\times(3\times n\times n)}$$

It is clear that $H^\top = H$ and that $H^2 = H$. Hence $\left(H_{mask}H_{mask}^\top + \frac{\sigma^2}{r_t^2}I\right)^{-1} H = \mathrm{Diag}\left(\frac{\delta_{c,i,j}}{1+\frac{\sigma^2}{r_t^2}}\right)_{c,i,j}$.

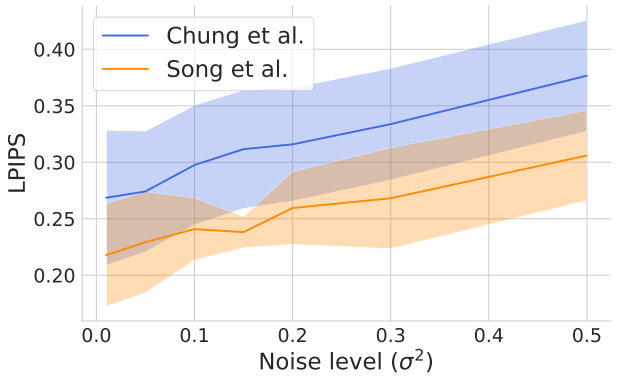
We conclude that the guidance term from equation (3) can be simply computed in the image space by using the mask M:

$$\left(\left(H_{mask}H_{mask}^\top + \frac{\sigma^2}{r_t^2}I\right)^{-1} H\right)^\top (y - H_{mask}\hat{x}_t) = \frac{1}{1+\frac{\sigma^2}{r_t^2}} M * (y - \hat{x}_t)$$

We perform the same type of experiments than for denoising, with 5 images and 7 levels of added Gaussian noise. We consider on top of this a 85×85 mask located between the center and the top right corner of the image. Examples of such faces with their masked and noisy versions, as well as their reconstructions using both Chung's and Song's methods (guiding factor of $\zeta = 0.1$), are shown in appendix (figure 4). Results in terms of LPIPS are given figure 2b: the pseudoinverse guidance consistently outperforms Chung's reconstruction for all levels of noise.



(a) Average PSNR (and standard deviation) over 5 denoised images with levels of noise between 0.01 and 0.5.



(b) Average LPIPS (and standard deviation) over 5 inpainted and denoised images (mask of size 85×85 and noise level between 0.01 and 0.5).

Figure 2: Comparison between Chung's method and the pseudoinverse guidance for denoising (left) and inpainting-denoising (right).

4.4 Uniform and Gaussian deblurring

The deblurring task is the most delicate we had to manage. Similar to the inpainting task, we need to formulate it as a multiplication of a vectorized image in by a matrix $H \in \mathbb{R}^{65536\times 65536}$ ($256 \times 256 = 65536$). However, the guidance expression (equation (3)) requires inverting $HH^\top + \frac{\sigma_t^2}{\nabla_t^2}I$, which entails a cubic computational complexity. Our initial attempts using sparse matrices were promising but turned out being unsuccessful.

We ultimately adopt an effective approach following Kawar et al. [7] in the DDRM (Denoising Diffusion Restoration Model) paper, which focuses on separable blurring with kernel $K = rc^\top$. This

formulation allows the blurred image to be obtained by manipulating rows and columns of the image independently. It is expressed as a matrix multiplication in the image domain: $A_c X A_r^\top$, where A_c and A_r apply a 1D convolution with kernel c and r. This is equivalent to applying Hx, where $H = A_r \otimes A_c$, x is the vectorised image in $\mathbb{R}^{65536}$ and $\otimes$ is the Kronecker product.

Then, an easy way to compute the inverse in equation (3) is to compute the SVD of H. By using $A_r = U_r \Sigma_r V_r^\top$ and $A_c = U_c \Sigma_c V_c^T$, we can compute it implicitly:

$$H = (U_r \otimes U_c)(\Sigma_r \otimes \Sigma_c)(V_r \otimes V_c)^\top = U \Sigma V^\top$$

where U, Σ, V have been permuted so that the singular values Σ are sorted.

Using this property allows us to simulate the multiplication of a vector by U, V, $V^\top$, etc., without having to store the matrix of dimension $\mathbb{R}^{65536 \times 65536}$. In the code we apply an operator H directly to the image and compute implicitly the vectorised computation Hx.

Here is how we reformulate the problem to perform the operation implicitly:

$$H^\top \left(\tilde{H}^{-1} \left(y - H(\hat{x_t}) \right) \right)$$

with

$$\tilde{H} = HH^\top + \frac{\sigma_t^2}{r_t^2} I = \left(U \left(\Sigma^2 + \frac{\sigma_t^2}{r_t^2} I \right) \right) U^\top = U \Sigma^2 U^\top + \sigma_t^2$$

$$\tilde{H}^{-1} = U \underbrace{\left(\Sigma^2 + \frac{\sigma_t^2}{r_t^2} I \right)^{-1}}_{\tilde{\Sigma}} U^\top \qquad \text{where} \quad (\tilde{\Sigma})_i = \frac{1}{(\Sigma)_i + \frac{\sigma_t^2}{r_t^2}} \quad \forall i$$

While Song and al. only uses uniform deblurring, we also experiment with Gaussian deblurring. For both cases, we use a scaling factor for the guidance of $\zeta = 0.04$. Unfortunately, our pseudoinverse guidance leads to highly unstable results: the generated images are often non-sense or even full black or full white. For this reason, we were not able to compute robust metrics on a subset of reconstructed images.

However, we present in the appendix an example of deblurred image for both the uniform and the gaussian blurring, with different levels of added noise (figure 5 and 6). Despite its obvious instability compared to Chung's method, the pseudoinverse guidance seems to outperform it in terms of quality. At least on those two images, we can indeed perceive much more resemblance between the faces reconstructed with Song than with Chung.

References

[1] Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations (ICLR)*. ICLR, 2023. Affiliations: KAIST and Los Alamos National Laboratory.

[2] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. Project page: `https://hojonathanho.github.io/diffusion/`.

[4] Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. Code: `https://github.com/openai/guided-diffusion/`.

[5] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.

[7] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, 2022.

A Equivalence between score and noise formulations of the problem

In the course session and in our implementation of the DDPM class, we use a network that approximates the noise and not the score function. Let's show that our pseudo-code is equivalent to the formulation used in Chung et al. and Song et al.

First let's make a correspondence between the notation of the articles and ours. Recall that

$$\alpha_t = 1 - \beta_t, \quad \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

Form 1 of Chung and al. (in terms of x_0 and x_t):

$$\tilde{\mu}\big(x_t, x_0\big) \;=\; \frac{\sqrt{\bar{\alpha}_{t-1}}\,\beta_t}{1-\bar{\alpha}_t}\,x_0 \;+\; \frac{\sqrt{\alpha_t}\,\big(1-\bar{\alpha}_{t-1}\big)}{1-\bar{\alpha}_t}\,x_t$$

Form 2 of our DDPM (in terms of x_t and the noise ϵ):

$$\tilde{\mu}\big(x_t, \epsilon\big) \;=\; \frac{1}{\sqrt{\alpha_t}}\Big(x_t \;-\; \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon\Big)$$

By definition of the forward diffusion,

$$x_t \;=\; \sqrt{\bar{\alpha}_t}\,x_0 \;+\; \sqrt{1-\bar{\alpha}_t}\,\epsilon, \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}).$$

So x_0 writes:

$$x_0 \;=\; \frac{1}{\sqrt{\bar{\alpha}_t}}\,x_t \;-\; \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\,\epsilon.$$

By splitting the factor $\sqrt{\bar{\alpha}_t} = \sqrt{\bar{\alpha}_{t-1}}\,\sqrt{\alpha_t}$,

$$\begin{aligned}\tilde{\mu} &= \frac{\sqrt{\overline{\alpha}_{t-1}}\,\beta_t}{1-\overline{\alpha}_t}\Big(\frac{x_t}{\sqrt{\overline{\alpha}_t}} \;-\; \frac{\sqrt{1-\overline{\alpha}_t}}{\sqrt{\overline{\alpha}_t}}\,\epsilon\Big) \;+\; \frac{\sqrt{\alpha_t}\,\big(1-\overline{\alpha}_{t-1}\big)}{1-\overline{\alpha}_t}\,x_t \\ &= \Big(\frac{1-\alpha_t}{\sqrt{\alpha_t}\,\big(1-\overline{\alpha}_t\big)} \;+\; \frac{\alpha_t\,\big(1-\overline{\alpha}_{t-1}\big)}{\sqrt{\alpha_t}\,\big(1-\overline{\alpha}_t\big)}\Big)\,x_t \;-\; \frac{\beta_t}{\sqrt{\alpha_t}\,\sqrt{1-\overline{\alpha}_t}}\,\epsilon \\ &= \frac{1}{\sqrt{\alpha_t}\,\big(1-\overline{\alpha}_t\big)}\Big[(1-\alpha_t) \;+\; \alpha_t\,\big(1-\overline{\alpha}_{t-1}\big)\Big]\,x_t \;-\; \frac{\beta_t}{\sqrt{\alpha_t}\,\sqrt{1-\overline{\alpha}_t}}\,\epsilon\end{aligned}$$

Since $\overline{\alpha}_t = \overline{\alpha}_{t-1}\,\alpha_t$, we have $(1-\alpha_t) \;+\; \alpha_t\,\big(1-\overline{\alpha}_{t-1}\big) \;=\; 1 \;-\; \alpha_t\,\overline{\alpha}_{t-1} \;=\; 1-\overline{\alpha}_t$. Therefore,

$$\tilde{\mu}\big(x_t,\epsilon\big) \;=\; \frac{1}{\sqrt{\alpha_t}}\Big(x_t \;-\; \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon\Big),$$

which proves the equivalence of form 1 and 2.

B Product of two Gaussian functions

If we note the product of two Gaussian functions can be expressed as:

$$f(x)g(x) = \frac{1}{2\pi\sigma_f\sigma_g}\exp\left(-\frac{(x-\mu_f)^2}{2\sigma_f^2} - \frac{(x-\mu_g)^2}{2\sigma_g^2}\right).$$

where μ_{fg} is defined as:

$$\mu_{fg} = \frac{\mu_f\sigma_g^2 + \mu_g\sigma_f^2}{\sigma_f^2+\sigma_g^2}$$

and σ_{fg}^2 is given by:

$$\sigma_{fg}^2 = \frac{\sigma_f^2\sigma_g^2}{\sigma_f^2+\sigma_g^2}.$$

After some derivations, the product can be rewritten as:

$$f(x)g(s) = \frac{1}{2\pi\sigma_f\sigma_g}\exp\left(-\frac{(x-\mu_{fg})^2}{2\sigma_{fg}^2}\right)\exp\left(\frac{(\mu_f-\mu_g)^2}{2(\sigma_f^2+\sigma_g^2)}\right)$$

which explains the formulas seen in Song et al.

C Reconstructed images for tasks of denoising, inpainting, uniform and gaussian deblurring

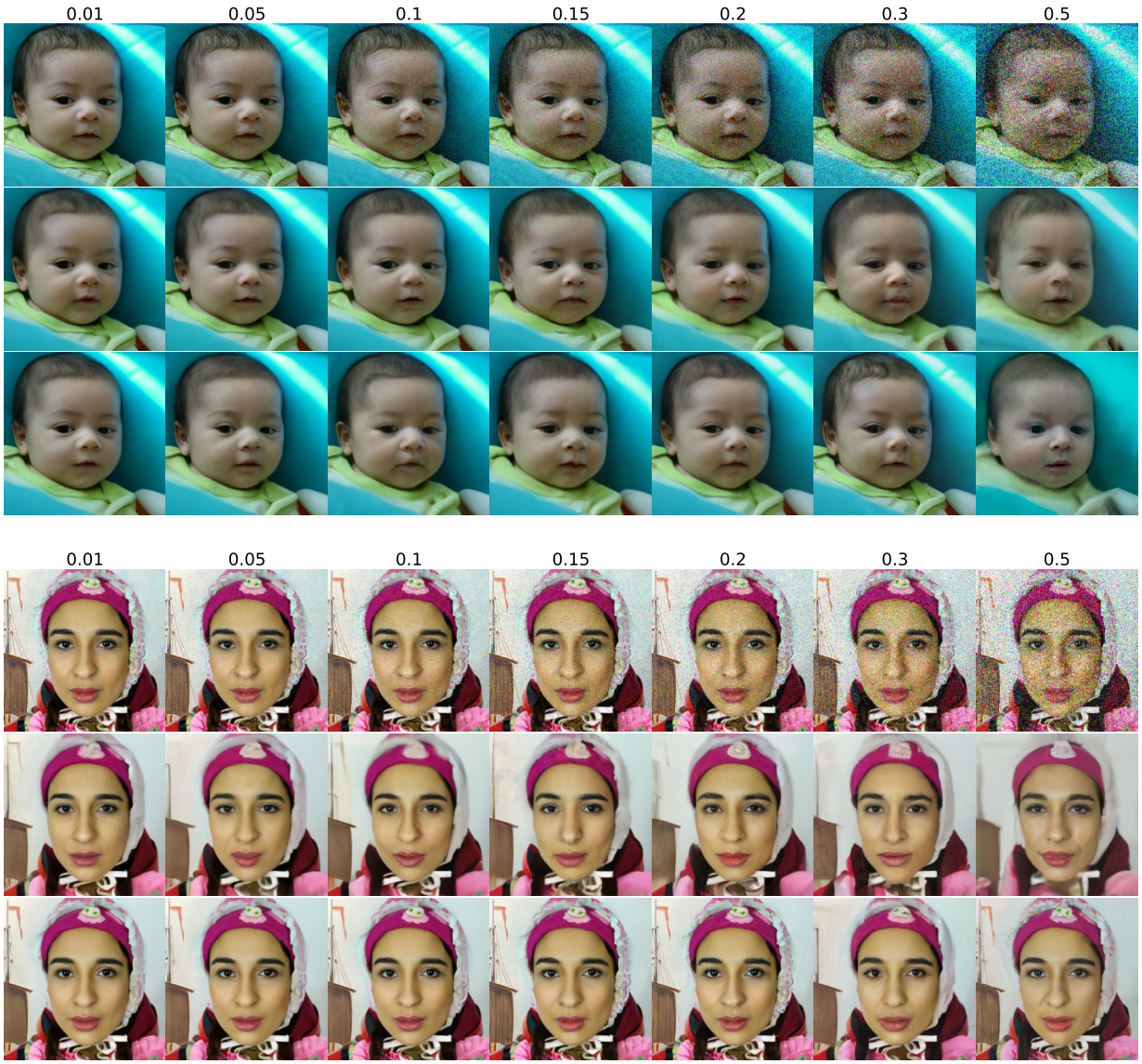


Figure 3: Denoising task for different levels of noise ranging from 0.01 to 0.5. The first line shows the noisy versions of the images, the second line their reconstructions using Chung's method, the third line their reconstructions using our pseudo-inverse guidance.

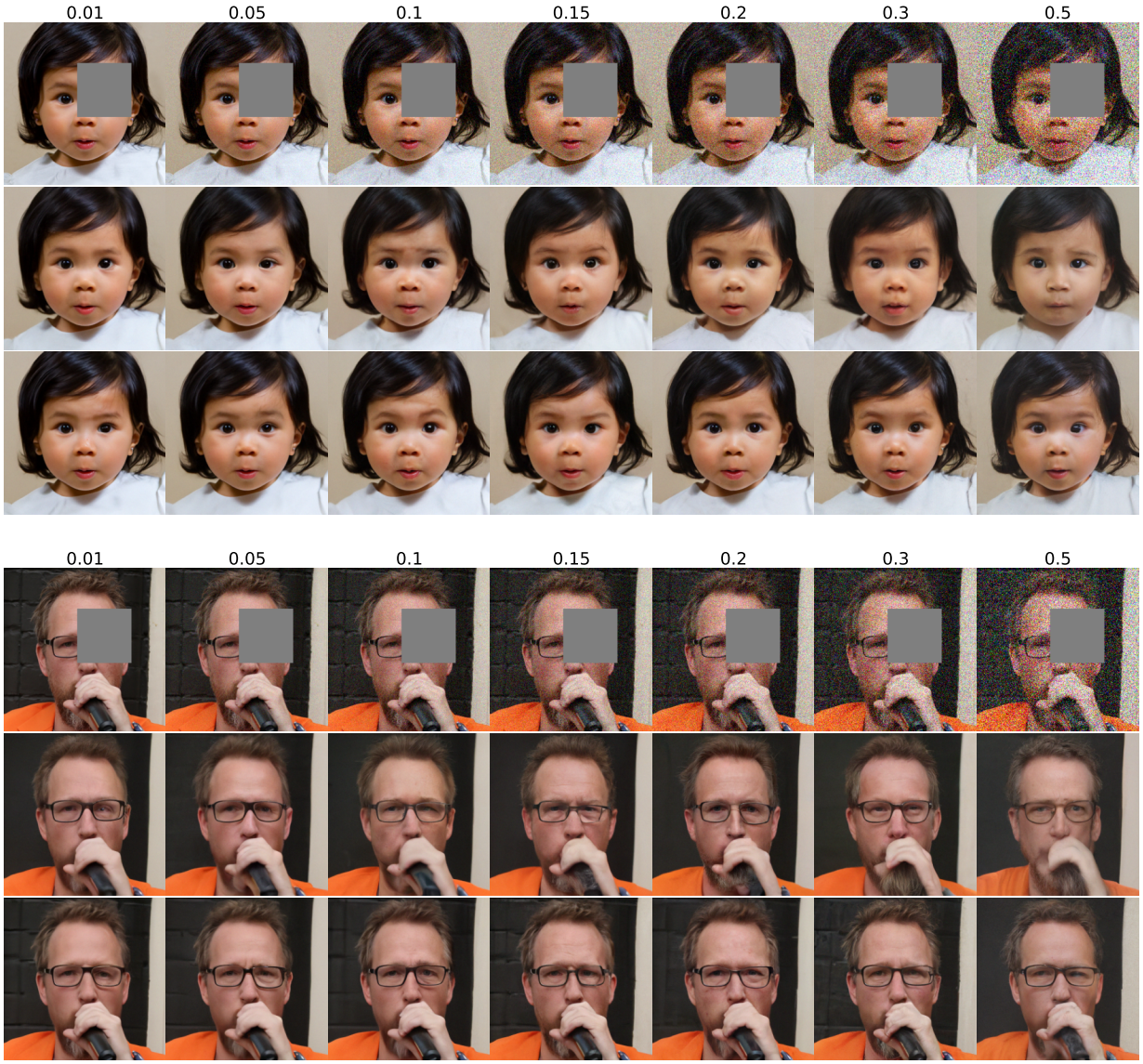


Figure 4: Inpainting task combined with denoising (for different levels of noise ranging from 0.01 to 0.5). The first line shows the masked and noisy versions of the images, the second line their reconstructions using Chung's method, the third line their reconstructions using our pseudo-inverse guidance.

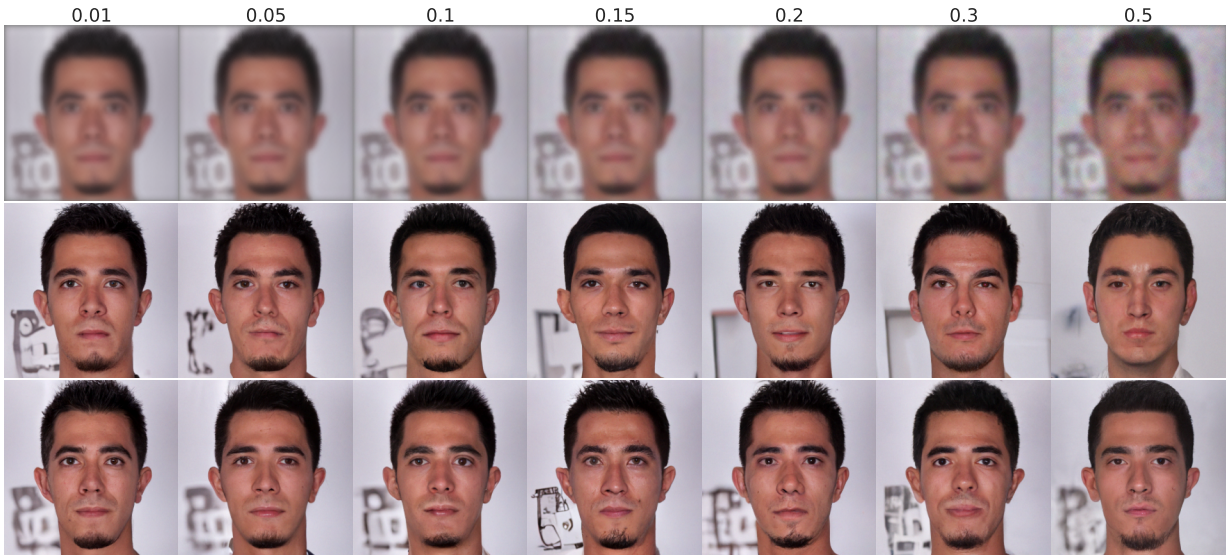


Figure 5: Uniform deblurring task combined with denoising (for different levels of noise ranging from 0.01 to 0.5). The first line shows the blurred and noisy versions of the images, the second line their reconstructions using Chung's method, the third line their reconstructions using our pseudo-inverse guidance.

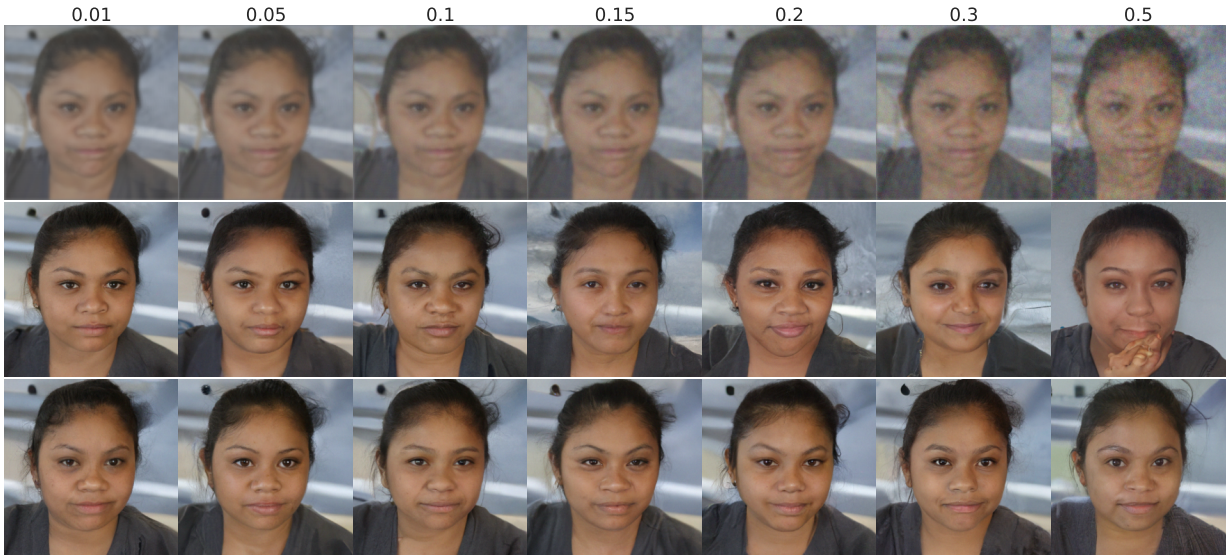


Figure 6: Gaussian deblurring task combined with denoising (for different levels of noise ranging from 0.01 to 0.5). The first line shows the blurred and noisy versions of the images, the second line their reconstructions using Chung's method, the third line their reconstructions using our pseudo-inverse guidance.